



Proceedings of the 13th International Conference on Educational Data Mining

Anna N. Rafferty, Jacob Whitehill, Cristobal Romero, Violetta Cavalli-Sforza
(eds).



Proceedings of the 13th International Conference on Educational Data Mining
Anna N. Rafferty, Jacob Whitehill, Cristobal Romero, Violetta Cavalli-Sforza (eds).
July 10th – 13th 2020
ISBN: 978-1-7336736-1-7

PREFACE

For this 13th iteration of the International Conference on Educational Data Mining (EDM 2020), the conference was held completely online. EDM is organized under the auspices of the International Educational Data Mining Society in Montreal, Canada. The conference, held July 10th through 13th, 2020, follows twelve previous editions (Montreal 2019, Buffalo 2018, Wuhan 2017, Raleigh 2016, Madrid 2015, London 2014, Memphis 2013, Chania 2012, Eindhoven 2011, Pittsburgh 2010, Cordoba 2009, and Montreal 2008).

The official theme of this year’s conference is Improving Learning Outcomes for All Learners. The theme comprises two parts: (1) Identifying actionable learning or teaching strategies that can be used to *improve* learning outcomes, not just predict them. (2) Using EDM to promote more *equitable* learning across diverse groups of learners, and to benefit underserved communities in particular. This year’s conference features three invited talks: Alina von Davier, Chief Officer at ACTNNext; Abelardo Pardo, Professor and Dean of Programs (Engineering), at UniSA STEM, University of South Australia; and Kobi Gal, Associate Professor at the Department of Software and Information Systems Engineering at Ben-Gurion University of the Negev, and Reader at the School of Informatics at the University of Edinburgh.

Building on the policy started in 2019, EDM 2020 used a double-blind review process. The conference’s Program Committee was also significantly expanded compared to 2019 in an effort to reduce the average load per reviewer and thereby increase the quality of reviews. This year we received a total of 98 full-paper submissions and 64 short-paper submissions. From the full-paper submissions, 30.6% were accepted as full papers, 16.3% were accepted as short papers, and 15.3% were accepted as posters. From the short-paper submissions, 21.9% were accepted as short papers and 23.4% were accepted as posters.

Review & Decision Processes: For transparency and possible benefit of future EDM conferences, we are providing a detailed description of the paper review and decision processes for the Full and Short paper tracks at EDM 2020:

1. After all papers were submitted, the Program Committee (PC) and Senior Program Committee (SPC) members bid on which papers they would like to review.
2. If committee members did not bid on papers after several reminders, bids were entered for them. This was done automatically via the EasyChair conference management system if the committee members had entered topics. Otherwise, one of the Program Chairs entered topics for the committee members based on examining publications in their Google Scholar profile; these topics were then used to automatically create bids.
3. Given the PC and SPC bids, the Program Chairs assigned papers to reviewers using EasyChair’s automatic assignment option. This assignment maximizes the total score of the assignment, with high weight on matches where the bid was a “yes”, medium weight on matches where the bid was a “maybe”, and low weight on matches where the bid was a “no”. Each paper was assigned to one SPC member and two PC members. Each PC member received at most 5 papers, and each SPC member received at most 4 papers. The automated reviewing assignment was manually checked to ensure fairness to reviewers in being primarily assigned papers for which they had entered positive bids, fairness to papers in being primarily assigned reviewers who had bid positively on that paper, and that automatic conflict detection had accurately detected conflicts. One set of changes involving three papers was made based on this

manual check due to assigning a paper to a reviewer that she had bid “no” on and that did not match her stated topics of expertise. In a separate change, another swap was made to prevent a reviewer from being assigned their own paper, as the authorship and reviewer information on EasyChair did not exactly match.

4. In an effort to increase the mean and decrease the variance in review quality, the Program Chairs defined reviewing guidelines, both for the PC and the SPC. These guidelines were posted to the EDM 2020 website and also linked in emails sent to reviewers.
5. At the end of the review period, the Program Chairs identified papers that received fewer than 3 reviews, as well as papers whose reviews were clearly lacking (e.g., just 1-2 sentences). Emergency reviewers (including the Program Chairs) were identified, and papers were assigned to them.
6. The Program Chairs examined the meta-reviews and acceptance/rejection recommendations for all papers. For any papers lacking a meta-review, the Program Chairs read the reviews and the paper, wrote a meta-review, and arrived at a recommendation for acceptance/rejection.
7. Papers were ranked by their unweighted average review scores. The Program Chairs then manually identified and examined papers in “critical regions” of the ranking in which there was large variance in the meta-reviewers’ decision recommendations (Accept as Full, Accept as Short, Accept as Poster, Reject). The goal here was to ensure that, in the opinions of both Program Chairs, all papers accepted as either Full or Short exhibited sufficient rigor for publication as such. When in doubt, the more conservative outcome (i.e., Accept as Short rather than Full, or Accept as Poster rather than Short) was chosen. In particular:
 - (a) For the Full paper track, the following range was calculated: Let m_f be the lowest score of any paper recommended by its meta-reviewer for “Accept as full”, and let n_s be the highest score of any paper recommended by its meta-reviewer for “Accept as short”. For any paper recommended for “Accept as full” whose score was in $[m_f, n_s]$, the Program Chairs discussed the paper and decided jointly whether to Accept as Full or Short. This deliberation focused on the question: “Do the reviewers point out important methodological or other fundamental problems that could significantly threaten validity?”
 - (b) The analogous process (both for papers submitted as Full, and for papers submitted as Short) was applied to papers whose unweighted average review scores were in the range $[m_s, n_p]$, where m_s is the lowest score of any paper recommended for Accept as Short and n_p is the highest score of any paper recommended for Accept as Poster.
 - (c) All other papers – i.e., those whose unweighted average review scores were outside of the ranges described above – were accepted/rejected according to the recommendation of their assigned meta-reviewer.

During all aspects of both the Review and Decision processes, no Program Chair examined or handled any paper on which he/she was a co-author; any such paper was seen and handled exclusively by the other Chair to avoid a conflict of interest. (No papers were co-authored by both Program Chairs.)

Note that papers submitted to the Industry, Doctoral Consortium, Poster/Demo, and Workshop components of EDM 2020 had their own reviewing processes that were defined by the corresponding chairs in consultation with the Program Chairs. Papers published in the Poster/Demo track are the union of those submitted & accepted as Posters/Demos, and those submitted to either the Full

or Short tracks that were accepted as Posters.

Posters/Demos: In addition to the Full or Short paper submissions that were accepted as posters mentioned above, there was a dedicated Poster/Demo track to which papers could be submitted directly. This track accepted 14 contributions out of 17 submissions.

JEDM: Together with the Journal of Educational Data Mining (JEDM), the EDM 2020 conference held a JEDM Track that provides researchers a venue to deliver more substantial mature work than is possible in a conference proceeding and to present their work to a live audience. The papers submitted to this track followed the JEDM peer review process. Two JEDM papers are featured in the conference’s program.

Industry: The main conference invited contributions to an Industry Track in addition to the main track. The EDM 2020 Industry Track received 6 submissions of which 5 were accepted.

Doctoral Consortium: The EDM conference continues its tradition of providing opportunities for young researchers to present their work and receive feedback from their peers and senior researchers. The doctoral consortium this year features 19 such presentations.

Paper Topics: In terms of topics of all submitted papers, the table below lists the most popular keywords associated with papers as selected by the authors themselves from a keyword list created by the Program Chairs:

Topic	# Paper Submissions
Log files/transaction logs	105
Modeling student learning	72
Other supervised machine learning	63
Post-secondary/College	55
Assessment	53
Intelligent tutoring systems	49
Natural language	48
Neural networks & deep learning	42
Unsupervised learning and clustering methods	41
Supporting teachers	33
MOOCs	30
K-12 classrooms	30
Building frameworks for EDM	30
Predicting attrition/drop-out	24
Data visualization methods	19
Informal learning environments	17
Collaborative learning	17
Images/video	17
Adult learning	17
Game-based learning	16
Multimodal analytics	15
Topic modeling	14
Closing the loop between research and practice	14
Advancing theories of learning	14
Building domain knowledge models	13
Bayesian models	12
Equity and fairness in EDM	10
Lab-based experiments	10
Socio-emotional learning and affect	8
Physiological sensors	8
Crowdsourcing	7
Social network analysis	6
Lifelong learning	6
Reinforcement learning	5
Treatment effect estimation	3
Causal inference techniques	2
Issues of Accessibility in Learning	1
Audio	1

Test of Time Award: Following in the footsteps of last year’s conference, EDM 2020 also includes an invited talk by the authors of the 2019 winner of the EDM Test of Time Award. This year’s talk is delivered by Ryan Baker and Kalina Yacef.

Workshops & Tutorials: In addition to the main program, there are workshops and tutorials on: Causal Inference in Educational Data Mining; Educational Data for Mining in Computer Science Education (CSDM); FATED: Fairness, Accountability, and Transparency in Educational Data (Mining); Reproducibility and Replication of Analytic Methods with LearnSphere; The Learner Data Institute: Big Data, Research Challenges, & Science Convergence in Educational Data Science; and An Introduction to Neural Networks.

Coronavirus: This year’s conference was originally arranged to take place in Ifrane, Morocco. Due to the SARS-CoV-2 (coronavirus) epidemic, EDM 2020, as well as most other academic conferences in 2020, had to be changed to a purely online format. This presented some difficulties, especially of how to engage and encourage interaction among participants using just Zoom and other online tools rather than face-to-face meetings. However, it also significantly reduced the costs of conducting and attending the conference since physical meeting spaces, air travel, and on-site lodging were no longer necessary – and this arguably increased our conference’s accessibility. To facilitate efficient transmission of presentations, especially given that not everyone’s Internet connection could be guaranteed to be stable, we required all paper presenters to pre-record their presentation as a video and then to host it on YouTube. Moreover, we asked that all presenters enable closed-captioning (CC), for the benefit of deaf people and those hard of hearing, as well as non-native English speakers who prefer to read than to listen to audio.

Thanks: We thank ACTNext as a sponsor of EDM 2020 for its generous support, especially during this financially difficult time of the coronavirus. We are also grateful to the individual conference chairs, the senior program committee, regular program committee members, sub-reviewers, emergency reviewers, and IEDMS board members, without whose expert input and hard work this conference would not be possible. Finally, we thank the entire organizing team and all authors who submitted their work to EDM 2020.

<i>Anna N. Rafferty</i>	Carleton College, USA	Program Chair
<i>Jacob Whitehill</i>	Worcester Polytechnic Institute, USA	Program Chair
<i>Cristobal Romero</i>	University of Cordoba, Spain,	General Chair
<i>Violetta Cavalli-Sforza</i>	Al Akhawayn University in Ifrane, Morocco	General Chair

June 23rd, 2020

ORGANIZING COMMITTEE

General Chairs: Cristobal Romero (University of Cordoba, Spain) and Violetta Cavalli-Sforza (Al Akhawayn University in Ifrane, Morocco)

Program Chairs: Anna N. Rafferty (Carleton College, USA) and Jacob Whitehill (Worcester Polytechnic Institute, USA)

Workshop & Tutorial Chairs: François Bouchet (Sorbonne University, France) and Vanda Lúengo (Sorbonne Université, France)

Industry Track Chairs: Shonte Stephenson (BrightBytes, USA) and Nigel Bosch (University of Illinois at Urbana-Champaign, USA)

Doctoral Consortium Chairs: Neil Heffernan (Worcester Polytechnic Institute, USA) and Carol Forsyth (Educational Testing Service, USA)

JEDM Track Chairs: Michel Desmarais (Polytechnique Montreal, Canada) and Agathe Merceron (Beuth University of Applied Sciences, Germany)

Poster & Demo Track Chairs: Juan Alfonso Lara Torralbo (UDIMA, Spain) and Rebeca Cerezo (University of Oviedo, Spain)

Publication/Proceedings Chairs: Pedro J. Muñoz-Merino (Universidad Carlos III de Madrid, Spain) and Alexandra I. Cristea (Durham University, UK)

Sponsorship Chairs: Piotr Mitros (ETS, USA) and Olga C. Santos (UNED, Spain)

Publicity/Social Media Chair: Miguel Á. Conde (University of Leon, Spain)

Web Chair: Paul Salvador Inventado (California State University, USA).

IEDMS Officers:

Kenneth Koedinger,	President	Carnegie Mellon University, USA
Mingyu Feng,	Treasurer	WestEd, USA

IEDMS Board of Directors:

Rakesh Agrawal	Data Insights Laboratories, USA
Ryan Baker	University of Pennsylvania, USA
Tiffany Barnes	North Carolina State University, USA
Michel Desmarais	Ecole Polytechnique de Montreal, Canada
Sidney D'Mello	University of Colorado Boulder, USA
Luc Paquette	University of Illinois Urbana-Champaign, USA
John Stamper	Carnegie Mellon University, USA
Mykola Pechenizkiy	Eindhoven University of Technology, Netherlands
Kalina Yacef	University of Sydney, Australia

Program Committee

Cecilia Aguerrebere	Duke University
Giora Alexandron	Weizmann Institute of Science
Ivon Arroyo	University of Massachusetts Amherst
Burcu Arslan	Educational Testing Service
Roger Azevedo	University of Central Florida
Costin Badica	University of Craiova, Computer and Information Technology Department, Romania
Tiffany Barnes	North Carolina State University
Gautam Biswas	Vanderbilt University
Nigel Bosch	University of Illinois Urbana-Champaign
Anthony F. Botelho	Worcester Polytechnic Institute
Jesus G. Boticario	UNED
François Bouchet	Sorbonne Université - LIP6
Yolaine Bourda	LRI, CentraleSupélec
Kristy Elizabeth Boyer	University of Florida
Javier Bravo-Agapito	Madrid Open University (UDIMA)
Keith Brawner	United States Army Research Laboratory
Armelle Brun	LORIA - Université Nancy 2
Renza Campagni	Università degli Studi di Firenze
Alberto Cano	Virginia Commonwealth University
Rebeca Cerezo	University of Oviedo
Guanliang Chen	Monash University
Min Chi	BeiKaZhouLi
Irene-Angelica Chounta	University of Tartu
Mihaela Cocea	School of Computing, University of Portsmouth
Chad Coleman	Teachers College - Columbia University
Miguel Ángel Conde	University of León
Alexandra Cristea	Durham University
Sidney D'Mello	University of Colorado Boulder
Hassane Darhmaoui	Al Akhawayn University in Ifrane
Michel Desmarais	Ecole Polytechnique de Montreal
Nicholas Diana	Carnegie Mellon University
Shayan Doroudi	Carnegie Mellon University
Kerrie Douglas	Purdue University
Michael Eagle	George Mason University
Bruno Emond	National Research Council Canada
Stephen Fancsali	Carnegie Learning, Inc.
Mingyu Feng	WestEd
Carol Forsyth	Educational Testing Service
Davide Fossati	Emory University
Carlos García-Martínez	Computing and Numerical Analysis Dept. Univ. of Córdoba
Joshua Gardner	University of Michigan
Dragan Gasevic	Monash University
Adam Gaweda	North Carolina State University
Niki Gitinabard	North Carolina State University

José González-Brenes	Chegg
Soumya Gosukonda	Chegg Inc.
Philip Guo	University of California San Diego
Jiangang Hao	Educational Testing Service
Erik Harpstead	Carnegie Mellon University
Fatima Harrak	LIP6 - Université Pierre et Marie Curie
Neil Heffernan	Worcester Polytechnic Institute
Erik Hemberg	Massachusetts Institute of Technology
Kenneth Holstein	CMU
Sharon Hsiao	Arizona State University
Paul Hur	University of Illinois at Urbana-Champaign
Stephen Hutt	University of Colorado Boulder
Paul Salvador Inventado	California State University Fullerton
Vladimir Ivančević	University of Novi Sad, Faculty of Technical Sciences
Jina Kang	Utah State University
Farzaneh Khoshnevisan	North Carolina State University
Ken Koedinger	Carnegie Mellon University
Irena Koprinska	The University of Sydney
Sotiris Kotsiantis	University of Patras
Sébastien Lallé	The University of British Columbia, Department of Computer Science
Juan Alfonso Lara Torralbo	UDIMA
James Lester	North Carolina State University
Chen Lin	North Carolina State University
Vanda Luengo	Sorbonne Université - LIP6
Ivan Luković	University of Novi Sad, Faculty of Technical Sciences
Maria Luque	University of Cordoba
Collin Lynch	North Carolina State University
Mirko Marras	University of Cagliari
Noboru Matsuda	North Carolina State University
Agathe Merceron	Beuth University of Applied Sciences Berlin
Donatella Merlini	Università di Firenze
Cristian Mihaescu	University of Craiova
Wookhee Min	North Carolina State University
Bradford Mott	North Carolina State University
Michael Mozer	Google Research
Tong Mu	Stanford University
Pedro J. Muñoz-Merino	Universidad Carlos III de Madrid
Roger Nkambou	Université du Québec À Montréal (UQAM)
Jaclyn Ocumpaugh	University of Pennsylvania
Andrew Olney	University of Memphis
Shai Olsher	University of Haifa
Korinn Ostrow	WPI
Una-May O'Reilly	Massachusetts Institute of Technology
Andreas Paepcke	Stanford University
Shalini Pandey	University of Minnesota
Luc Paquette	University of Illinois at Urbana-Champaign

Abelardo Pardo	University of South Australia
Zach Pardos	University of California, Berkeley
Philip I. Pavlik Jr.	University of Memphis
Mykola Pechenizkiy	Eindhoven University of Technology
Chris Piech	Stanford
Niels Pinkwart	Humboldt-Universität zu Berlin
Paul Stefan Popescu	University of Craiova
Thomas Price	North Carolina State University
Anna N. Rafferty	Carleton College
Martina Rau	University of Wisconsin - Madison, Department of Educational Psychology
Justin Reich	Massachusetts Institute of Technology
Steven Ritter	Carnegie Learning, Inc.
Ma. Mercedes T. Rodrigo	Department of Information Systems and Computer Science, Ateneo de Manila University
José Raúl Romero	University of Cordoba
Vasile Rus	The University of Memphis
Shaghayegh Sahebi	University at Albany - SUNY
Maria Ofelia San Pedro	ACT, Inc.
Olga C. Santos	aDeNu Research Group (UNED)
Shitian Shen	North Carolina State University
Vanessa Simmering	ACT, Inc.
Stefan Slater	Teachers College
Andy Smith	North Carolina State University
Marcus Specht	Delft University of Technology and Open University of the Netherlands
John Stamper	Carnegie Mellon University
Shonte Stephenson	BrightBytes
Angela Stewart	University of Colorado Boulder
Jun-Ming Su	Department of Information and Learning Technology, National University of Tainan
Ling Tan	Australian Council for Educational Research
Caitlin Tenison	Soar Technology
Stefan Trausan-Matu	University Politehnica of Bucharest
Jennifer Tsan	North Carolina State University
Rémi Venant	Le Mans Université - LIUM
Sebastián Ventura	University of Cordoba. Dept. of Computer Science and Numerical Analysis
Jill-Jênn Vie	Inria Lille
Elle Yuan Wang	Arizona State University
Stephan Weibelzahl	Private University of Applied Sciences Göttingen
Daniel Weitekamp	Carnegie Mellon University
Jacob Whitehill	Worcester Polytechnic Institute
Beverly Park Woolf	University of Massachusetts
Yiqiao Xu	North Carolina State University
Linting Xue	North Carolina State University
Amelia Zafra Gómez	Department of Computer Sciences and Numerical Analysis

Alfredo Zapata González
Diego Zapata-Rivera
Guojing Zhou
Craig Zilles
Amal Zouaq

Universidad Autonoma de Yucatan
Educational Testing Service
North Carolina State University
University of Illinois at Urbana-Champaign
Ecole Polytechnique de Montréal

SPONSORS



Best Paper Selection

Nominees for Best Paper Award were all full papers with an average review score greater than 2, where the average review score was weighted by the reviewer’s self-reported confidence. These papers were sent to and evaluated by all the board members of the International Educational Data Mining Society (IEDMS) who were not co-authors of one of the nominated papers. These board members selected the recipient of the Best Paper and Best Student Paper awards.

Best Paper Nominees: The nominated papers were:

1. Mike Wu, Richard Davis, Benjamin Domingue, Chris Piech and Noah Goodman. “Variational Item Response Theory: Fast, Accurate, and Expressive.”
2. Nigel Bosch, Wes Crues, Najmuddin Shaik and Luc Paquette. “Hello, [REDACTED]’: Protecting Student Privacy in Analyses of Online Discussion Forums.”
3. Adam Sales and John Pane. “The effect of teachers reassigning students to new Cognitive Tutor sections.”
4. Nathan Henderson, Vikram Kumara, Wookhee Min, Bradford Mott, Ziwei Wu, Danielle Boulden, Trudi Lord, Frieda Reichsman, Chad Dorsey, Eric Wiebe and James Lester. “Enhancing Student Competency Models for Game-Based Learning with a Hybrid Stealth Assessment Framework.”

Note that papers #1 and #4 above were also nominated for the **Best Student Paper Award**.

TABLE OF CONTENTS

Keynotes

Online Collaborative Student Group Learning.....	1
--	---

Kobi Gal

Contextualising Data Mining within Educational Experiences	2
--	---

Abelardo Pardo

An AI-enabled Ecosystem for Learning and Assessment	3
---	---

Alina Von Davier

JEDM Presentations

When is Deep Learning the Best Approach to Knowledge Tracing?	4
---	---

Theophile Gervet, Ken Koedinger, Jeff Schneider and Tom Mitchell

Who's learning? Using demographics in EDM research	5
--	---

Luc Paquette, Alexandra Andres, Jaclyn Ocumpaugh, Ryan Baker and Ziyue Li

Full Papers

Analyzing Student Strategies In Blended Courses Using Clickstream Data.....	6
---	---

Nil-Jana Akpinar, Aaditya Ramdas and Umut Acar

Unsupervised Approach for Modeling Content Structures of MOOCs	18
--	----

Fareedah Alsaad and Abdussalam Alawini

Increasing Enrollment by Optimizing Scholarship Allocations Using Machine Learning and Genetic Algorithms	29
---	----

Lovenoor Aulck, Dev Nambi and Jevin West

"Hello, [REDACTED]": Protecting Student Privacy in Analyses of Online Discussion Forums ..	39
--	----

Nigel Bosch, Wes Crues, Najmuddin Shaik and Luc Paquette

Predicting Engagement in Video Lectures	50
---	----

Sahan Bulathwela, María Pérez Ortiz, Aldo Lipani, Emine Yilmaz and John Shawe-Taylor

The Ebb and Flow of Student Engagement: Measuring motivation through temporal pattern of self-regulation.....	61
<i>Steven Dang and Kenneth Koedinger</i>	
Can we Take Advantage of Time-Interval Pattern Mining to Model Students Activity?	69
<i>Oriane Dermay and Armelle Brun</i>	
Automatic Subject-based Contextualisation of Programming Assignment Lists	81
<i>Samuel Fonseca, Filipe Dwan Pereira, Elaine H. T. Oliveira, David Fernandes, Leandro Carvalho and Alexandra Cristea</i>	
Enhancing Student Competency Models for Game-Based Learning with a Hybrid Stealth Assessment Framework	92
<i>Nathan Henderson, Vikram Kumara, Wookhee Min, Bradford Mott, Ziwei Wu, Danielle Boulden, Trudi Lord, Frieda Reichsman, Chad Dorsey, Eric Wiebe and James Lester</i>	
Harbingers of Collaboration? The Role of Early-Class Behaviors in Predicting Collaborative Problem Solving.....	104
<i>Paul Hur, Nigel Bosch, Luc Paquette and Emma Mercier</i>	
Evaluating sources of course information and models of representation on a variety of institutional prediction tasks.....	115
<i>Weijie Jiang and Zachary Pardos</i>	
Pick the Moment: Identifying Critical Pedagogical Decisions Using Long-Short Term Rewards.	126
<i>Song Ju, Guojing Zhou, Tiffany Barnes and Min Chi</i>	
Identifying At-Risk K-12 Students in Multimodal Online Environments: A Machine Learning Approach.....	137
<i>Hang Li, Wenbiao Ding and Zitao Liu</i>	
Erroneous Answers Categorization for Sketching Questions in Spatial Visualization Training...	148
<i>Tiffany Wenting Li and Luc Paquette</i>	
Getting too personal(ized): The importance of feature choice in online adaptive algorithms....	159
<i>Zhaobin Li, Luna Yee, Nathaniel Sauerberg, Irene Sakson, Joseph Jay Williams and Anna Rafferty</i>	

What Time is It? Student Modeling Needs to Know	171
<i>Ye Mao, Samiha Marwan, Thomas Price, Tiffany Barnes and Min Chi</i>	
Towards Suggesting Actionable Interventions for Wheel Spinning Students	183
<i>Tong Mu, Andrea Jetten and Emma Brunskill</i>	
Exploring homophily in demographics and academic performance using spatial-temporal student networks	194
<i>Quan Nguyen, Oleksandra Poquet, Christopher Brooks and Warren Li</i>	
The effect of teachers reassigning students to new Cognitive Tutor sections.....	202
<i>Adam Sales and John Pane</i>	
Feature Selection Metrics: Similarities, Differences, and Characteristics of the Selected Models	212
<i>Debopam Sanyal, Nigel Bosch and Luc Paquette</i>	
Learning a Policy Primes Quality Control: Towards Evidence-Based Automation of Learning Engineering	224
<i>Machi Shimmei and Noboru Matsuda</i>	
Recommending Remedial Readings Using Student's Knowledge state.....	233
<i>Khushboo Thaker, Lei Zhang, Daqing He and Peter Brusilovsky</i>	
Image Reconstruction of Tablet Front Camera Recordings in Educational Settings.....	245
<i>Rafael Wampfler, Andreas Emch, Barbara Solenthaler and Markus Gross</i>	
Variational Item Response Theory: Fast, Accurate, and Expressive.....	257
<i>Mike Wu, Richard Davis, Benjamin Domingue, Chris Piech and Noah Goodman</i>	
Student Subtyping via EM-Inverse Reinforcement Learning	269
<i>Xi Yang, Guojing Zhou, Michelle Taub, Roger Azevedo and Min Chi</i>	
Analyzing Student Procrastination in MOOCs: A Multivariate Hawkes Approach	280
<i>Mengfan Yao, Shaghayegh Sahebi and Reza Feyzi Behnagh</i>	

Towards Accurate and Fair Prediction of College Success: Evaluating Different Sources of Student Data	292
<i>Renzhe Yu, Qiujie Li, Christian Fischer, Shayan Doroudi and Di Xu</i>	
The NAEP EDM Competition: Theory-Driven Psychometrics and Machine Learning for Predictions Based on Log Data	302
<i>Fabian Zehner, Scott Harrison, Beate Eichmann, Tobias Deribo, Daniel Bengs, Nico Andersen and Carolin Hahnel</i>	
Modeling Knowledge Acquisition from Multiple Learning Resource Types	313
<i>Siqian Zhao, Chunpai Wang and Shaghayegh Sahebi</i>	
Predicting Student Performance in a Master's Program in Data Science using Admissions Data	325
<i>Yijun Zhao, Qiangwen Xu, Ming Chen and Gary Weiss</i>	
Short Papers	
Decomposition of Response Time to Give Better Predictions of Children's Reading Comprehension	334
<i>Zhila Aghajari, Deniz Sonmez Unal, Mesut Erhan Unal, Ligia Gómez and Erin Walker</i>	
Whose Truth is the "Ground Truth"? College Admissions Essays and Bias in Word Vector Evaluation Methods	342
<i>Noah Arthurs and Aj Alvero</i>	
A Dataset of Learnersourced Explanations from an Online Peer Instruction Environment	350
<i>Sameer Bhatnagar, Michel Desmarais, Amal Zouaq and Elizabeth Charles</i>	
Effective Forum Curation via Multi-task Learning	356
<i>Faeze Brahman, Nikhil Varghese, Suma Bhat and Snigdha Chaturvedi</i>	
CSCLRec: Personalized Recommendation of Forum Posts to Support Socio-collaborative Learning	364
<i>Zhaorui Chen and Carrie Demmans Epp</i>	
Deep Embeddings of Contextual Assessment Data for Improving Performance Prediction	374
<i>Benjamin Clavié and Kobi Gal</i>	

More Data and Better Keywords Imply Better Educational Transcript Classification?	381
<i>Theodora Danciulescu, Stella Heras, Javier Palanca, Vicente Julian and Cristian Mihaescu</i>	
Zero-shot Learning of Hint Policy via Reinforcement Learning and Program Synthesis	388
<i>Aleksandr Efremov, Ahana Ghosh and Adish Singla</i>	
Investigating Relations between Self-Regulated Reading Behaviors and Science Question Difficulty	395
<i>Effat Farhana, Teomara Rutherford and Collin Lynch</i>	
Are You Really a Team Player?: Profiles of collaborative problem solvers in an online environment	403
<i>Carol Forsyth, Jessica Andrews-Todd and Jonathan Steinberg</i>	
Student Teamwork on Programming Projects. What can GitHub logs show us?	409
<i>Niki Gitinabard, Ruth Okoilu Akintunde, Yiqiao Xu, Sarah Heckman, Tiffany Barnes and Collin Lynch</i>	
Using Process Data to Evaluate Scientific Inquiry Practice in Technology-Enhanced Assessment	417
<i>Tao Gong, Lan Shuai, Burcu Arslan and Yang Jiang</i>	
Confident Learning Curves in Additive Factor Modeling	424
<i>Cyril Goutte and Guillaume Durand</i>	
Towards Fair Educational Data Mining: A Case Study on Detecting At-risk Students	431
<i>Qian Hu and Huzefa Rangwala</i>	
Using online text books and in-class quizzes to predict in class performance	438
<i>Noah Hunt-Isaak, Peter Cherniavsky, Mark Snyder and Huzefa Rangwala</i>	
Online Academic Course Performance Prediction using Relational Graph Convolutional Neural Network	444
<i>Hamid Karimi, Tyler Derr, Jiangtao Huang and Jiliang Tang</i>	
EDM and Privacy: Ethics and Legalities of Data Collection, Usage, and Storage	451
<i>Mark Klose, Vasvi Desai, Yang Song and Edward Gehringer</i>	

Course Recommendation for University Environment	460
<i>Boxuan Ma, Yuta Taniguchi and Shinichi Konomi</i>	
Methodology to measure of similarity in student video sequence of interactions.	467
<i>Boniface Mbouzao, Michel Desmarais and Ian Shrier</i>	
PIPE: Predicting Logical Programming Errors in Programming Exercises	473
<i>Dezhuang Miao, Yu Dong and Xuesong Lu</i>	
Incidence of teacher curricular emphasis in reading achievement of Uruguayan ninth-grade students	480
<i>Leonardo Moreno, Matias Núñez, Cecilia Emery, Inés Méndez, Elisa Borba and Eliana Lucián</i>	
Moving beyond Test Scores: Analyzing the Effectiveness of a Digital Learning Game through Learning Analytics	487
<i>Huy Nguyen, Xinying Hou, John Stamper and Bruce McLaren</i>	
Measuring Ability-to-Learn Using Parametric Learning Gain Functions.....	496
<i>Chris Piech, Engin Bumbacher and Richard Davis</i>	
Iterative Feature Engineering Through Text Replays of Model Error	503
<i>Stefan Slater, Ryan Baker and Yeyu Wang</i>	
Course Recommender Systems with Statistical Confidence	509
<i>Zachary Warnes and Evgueni Smirnov</i>	
Problem detection in peer assessments between subjects by effective transfer learning and active learning.....	516
<i>Yunkai Xiao, Gabriel Zingle, Qinjin Jia, Shoaib Akbar, Yang Song, Muyao Dong, Li Qi and Edward Gehringer</i>	
Dynamic knowledge embedding and tracing.....	524
<i>Liangbei Xu and Mark Davenport</i>	
Incorporating Task-specific Features into Deep Models to Classify Argument Components	531
<i>Linting Xue and Collin Lynch</i>	

A Quantitative Machine Learning Approach to Master Students Admission for Professional Institutions.....	538
--	-----

Yijun Zhao, Bryan Lackaye, Jennifer Dy and Carla Brodley

Posters

Where to aim? Factors that influence the performance of Brazilian secondary schools	545
---	-----

Paulo Jorge L. Adeodato and Rogerio Luiz C. S. Filho

A Procrastination Index for Online Learning Based on Assignment Start Time	550
--	-----

Lalitha Agnihotri, Ryan Baker and Steve Stalzer

Automated Assessment of Computer Science Competencies from Student Programs with Gaussian Process Regression.....	555
---	-----

Bita Akram, Hamoon Azizsoltani, Wookhee Min, Eric Wiebe, Anam Navied, Bradford Mott, Kristy Elizabeth Boyer and James Lester

Educational Data Mining and Personalized Support in Online Introductory Physics Courses...	561
--	-----

Farook Al-Shamali, Hongxin Yan, Sabine Graf and Fuhua Lin

First Steps Towards NLP-based Formative Feedback to Improve Scientific Writing in Hebrew..	565
--	-----

Moriah Ariely, Tanya Nazaretsky and Giora Alexandron

Discovering the Prerequisite Relationships Among Instructional Videos From Subtitles	569
--	-----

Mehmet Cem Aytekin, Stefan Rübiger and Yucel Saygin

A method for generating features representing the students' degree of anticipation/delay to complete assignments.....	574
---	-----

Francisco Cervera and Juan Lara

Predicting students' performance using emotion detection from face-recording video when interacting with an ITS.....	578
--	-----

Wilson Gustavo Chango Sailema, Miguel Sánchez, Rebeca Cerezo and Cristóbal Romero

Applying Recent Innovations from NLP to MOOC Student Course Trajectory Modeling.....	581
--	-----

Clarence Chen and Zachary Pardos

Return of the Student: Predicting Re-Engagement in Mobile Learning	586
<i>Maximillian Chen and Rene Kizilcec</i>	
Does autonomy help Help? The impact of unsolicited hints and choice on help avoidance and learning	591
<i>Christa Cody, Mehak Maniktala, David Warren, Min Chi and Tiffany Barnes</i>	
An EDM-based Multimodal Method for Assessing Learners' Affective States in Collaborative Crisis Management Serious Games	596
<i>Ibtissem Daoudi, Erwan Tranvouez, Raoudha Chebil, Bernard Espinasse and Wided Lejouad Chaari</i>	
Exploration of Process Mining Opportunities In Educational Software Engineering - The GitLab Analyser	601
<i>Philipp Dumbach, Alexander Aly, Markus Zrenner and Bjoern M. Eskofier</i>	
Comparing and combining tests for plagiarism detection in online exams	605
<i>Edward Gehringer, Xiaohan Liu, Abhirav Kariya and Guoyi Wang</i>	
Curriculum profile: modelling the gaps between curriculum and the job market	610
<i>Aleksandr Gromov, Andrei Maslennikov, Nikolas Dawson, Katarzyna Musial and Kirsty Kitto</i>	
Assessing Student Contributions in Wiki-based Collaborative Writing System	615
<i>Tianyu Hu, Guangzhong Sun and Zhongtian Xu</i>	
EAnalyst: Toward Understanding Large-scale Educational Data	620
<i>Tao Huang, Zhi Li, Hao Zhang, Huali Yang and Hekun Xie</i>	
How we talk about math: Leveraging naturalistic datasets to define the discourse of math in contrast to other domains	624
<i>Rachel Jansen and Ruthe Foushee</i>	
Predicting Student Dropout by Mining Advisor Notes	629
<i>J.D Jayaraman</i>	
Data Efficient Educational Assessment via Multi-Dimensional Pairwise Comparisons	633
<i>Hyunbin Loh, Piljae Chae and Chanyou Hwang</i>	

Using Edit Distance Trails to Analyze Path Solutions of Parsons Puzzles	638
<i>Salil Maharjan and Amruth Kumar</i>	
How Does Student Behaviour Change Approaching Dropout? A Study of Gender and School Year Differences	643
<i>Jessica Mcbroom, Irena Koprinska and Kalina Yacef</i>	
Measuring task difficulty for online learning environments where multiple attempts are allowed – the Elo rating algorithm approach	648
<i>Maciej Pankiewicz</i>	
Social Media Mining to Understand the Impact of Co-operative Education on Mental Health ..	653
<i>Mohammad S. Parsa and Lukasz Golab</i>	
Towards Temporality-Sensitive Recurrent Neural Networks through Enriched Traces	658
<i>Thomas Sergent, François Bouchet and Thibault Carron</i>	
Predicting and Understanding Success in an Innovation-Based Learning Course	662
<i>Lauren Singelmann, Enrique Alvarez, Ellen Swartz, Ryan Striker, Mary Pearson and Dan Ewert</i>	
Linguistic Changes across Different User Roles in MOOCs: What do they tell us?	667
<i>Lavendini Sivaneasharajah, Katrina Falkner and Thushari Atapattu</i>	
Toward a deep convolutional LSTM for eye gaze spatiotemporal data sequence classification ...	672
<i>Komi Sodoke, Roger Nkambou, Aude Dufresne and Issam Tanoubi</i>	
qDKT: Question-centric Deep Knowledge Tracing	677
<i>Shashank Sonkar, Andrew Lan, Andrew Waters, Phillip Grimaldi and Richard Baraniuk</i>	
IntelliMOOC: Intelligent Online Learning Framework for MOOC Platforms	682
<i>Patara Trirat, Sakonporn Noree and Mun Yong Yi</i>	
Using Association Rule Mining to Uncover Rarely Occurring Relationships in Two University Online STEM Courses: A Comparative Analysis	686
<i>Hannah Valdiviejas and Nigel Bosch</i>	

Claim Detection and Relationship with Writing Quality	691
<i>Qian Wan, Scott Crossley, Laura Allen and Danielle McNamara</i>	
VarFA: A Variational Factor Analysis Framework For Efficient Bayesian Learning Analytics ...	696
<i>Zichao Wang, Yi Gu, Andrew Lan and Richard Baraniuk</i>	
Next-Term Grade Prediction: A Machine Learning Approach	700
<i>Audrey Tedja Widjaja, Lei Wang, Nghia Trong Truong, Aldy Gunawan and Ee-Peng Lim</i>	
Detecting Problem Statements in Peer Assessments	704
<i>Yunkai Xiao, Gabriel Zingle, Qinjin Jia, Harsh Shah, Yi Zhang, Tianyi Li, Mohsin Karovaliya, Weixiang Zhao, Yang Song, Jie Ji, Ashwin Balasubramaniam, Harshit Patel, Priyanka Bhalasubramanian, Vikram Patel and Edward Gehringer</i>	
An Empirical Analysis of Skewed Temporal Data for Distribution-based Course Similarity	710
<i>Tao Xie, Chaohua Gong and Geping Liu</i>	
Semi-supervised Learning Method for Adjusting Biased Item Difficulty Estimates Caused by Nonignorable Missingness under 2PL-IRT Model	715
<i>Kang Xue</i>	
An effect-size-based temporal interestingness metric for sequential pattern mining	720
<i>Yingbin Zhang and Luc Paquette</i>	
Industry Track	
Dynamic knowledge tracing through data driven recency weights	725
<i>Deepak Agarwal, Ryan Baker and Anupama Muraleedharan</i>	
Auto generation of diagnostic assessments and their quality evaluation	730
<i>Soma Dhavala, Chirag Bhatia, Joy Bose, Keyur Faldu and Aditi Avasthi</i>	
Differential Responses to Personalized Learning Recommendations Revealed by Event-Related Analysis	736
<i>Kevin Dieter, Jamie Studwell and Kirk Vanacore</i>	

Prescribing Deep Attentive Score Prediction Attracts Improved Student Engagement	743
<i>YOUNGNAM LEE, BYUNGSOO KIM, DONGMIN SHIN, JUNGHOO KIM, JINEON BAEK, JINHWAN LEE and YOUNGDUCK CHOI</i>	
The Results of Zone of Proximal Development on Learning Outcome	749
<i>SHENGNI WANG, YUXIN ZHAO, WEI MA, ZHENJUN MA and RYAN BAKER</i>	
Doctoral Consortium	
Mutual spontaneous aid between students in distance learning and the role of the feeling of social belonging to a training group	754
<i>DALILA BEBBOUCHI</i>	
Structural Explanation of Automated Essay Scoring	758
<i>AFRIZAL DOEWES and MYKOLA PECHENIZKIY</i>	
Natural Language Processing for Open Ended Questions in Mathematics within Intelligent Tutoring Systems	762
<i>JOHN ERICKSON</i>	
Self-Regulated Learning and Science Reading of Middle-School Students	766
<i>EFFAT FARHANA, TEOMARA RUTHERFORD, and COLLIN LYNCH</i>	
Developing Curriculum Analytics and Student Social Networking for Graduate Employability Model	770
<i>ALEKSANDR GROMOV</i>	
Overcoming Foreign Language Anxiety in an Emotionally Intelligent Tutoring System	773
<i>DANEIH ISMAIL</i>	
The Effect of Visual Cues on Cognitive Load Depending on Self-Regulation in Video-Based Learning	776
<i>KAKYEONG KIM and IL-HYUN JO</i>	
Towards Understanding the Impact of Real-Time AI-Powered Educational Dashboards (RAED) on Providing Guidance to Instructors	781
<i>AJAY KULKARNI and MICHAEL EAGLE</i>	

Estimation for cognitive load in Video-based learning through Physiological Data and Subjective Measurement by Video Annotation	785
---	-----

In-Hye Lee

Exploration Maps, Beyond Top Scores: Designing Formative Feedback for Open-Ended Problems	790
---	-----

Aditi Mallavarapu and Leilah Lyons

Extending the Hint Factory: Towards Modelling Productivity for Open-ended Problem-solving	796
---	-----

Mehak Maniktala, Tiffany Barnes and Min Chi

Scalability in Online Computer Programming Education: Automated Techniques for Feedback, Evaluation and Equity	802
--	-----

Jessica Mcbroom, Kalina Yacef and Irena Koprinska

Investigating Students' Learning in Online Learning Environment	806
---	-----

Lavendini Sivaneasharajah, Katrina Falkner and Thushari Atapattu

Building Test Recommender Systems for e-Learning Systems	810
--	-----

Oana Maria Teodorescu

Crowd-sourcing and Automatic Generation of Semantic Information in Blended-Learning Environments	815
--	-----

Elad Yacobson

Workshop Abstracts

The Learner Data Institute: Big Data, Research Challenges, and Science Convergence in Educational Data Science	818
--	-----

Vasile Rus and Stephen Fancsali

An Introduction to Neural Networks	821
--	-----

Agathe Merceron and Ange Adrienne Nyamen Tato

Reproducibility and Replication of Analytic Methods with LearnSphere	824
--	-----

John Stamper, Kenneth Koedinger and Philip I. Pavlik Jr.

Educational Data Mining in Computer Science Education (CSEDM) Workshop	826
<i>Thomas Price, Peter Brusilovsky, Sharon Hsiao, Kenneth Koedinger and Yang Shi</i>	
Workshop: Causal Inference in Educational Data Mining	829
<i>Adam Sales</i>	
FATED: Fairness, Accountability, and Transparency in Educational Data (Mining)	831
<i>Nigel Bosch, Christopher Brooks, Shayan Doroudi, Josh Gardner, Kenneth Holstein, Renzhe Yu, Vie Jill-Jënn, Andrew Lan, Collin Lynch, Beverly Park Woolf, Mykola Pechenizkiy and Steven Ritter</i>	

Online Collaborative Student Group Learning

Keynote

Kobi Gal
Ben-Gurion University of the Negev, and University of Edinburgh

Abstract

Collaborative student learning has been shown to lead to significant academic benefits among students, and to improved social skills that are critical for the workforce, such as communication and teamwork. However, these benefits were limited to small face-to-face groups and required the support of human experts who actively monitored and guided the group's learning.

Technological advances now enable globally dispersed teams to collaborate online, from Q&A forums to virtual laboratories. Augmenting these settings with AI technology can scale up the benefits of collaborative group learning to online groups.

I will describe challenges to EDM research for supporting this new type of online teamwork, as well as opportunities for combining AI and learning analytics towards supporting students' learning and teachers' understanding of how students learn.

Biography

Kobi Gal is an Associate Professor at the Department of Software and Information Systems Engineering at Ben-Gurion University of the Negev, and Reader at the School of Informatics at the University of Edinburgh. Gal's work combines artificial intelligence algorithms with educational technology towards supporting students in their learning and teachers in their understanding how students learn. He has published widely in highly refereed venues on topics ranging from artificial intelligence to the learning and cognitive sciences.

Gal is the recipient of the Wolf foundation's 2013 Krill prize for young Israeli scientists, a Marie Curie International fellowship, and a three-time recipient of Harvard University's outstanding teacher award. He has received best paper awards at ACM Conference on User Modeling Adaptation and Personalization 2019 (UMAP-19), ACM conference on Economics and Computation 2016 (EC-16), Educational Data Mining 2014 (EDM-14). Gal is the acting president of the Israeli Association for Artificial Intelligence.

Kobi Gal "Online Collaborative Student Group Learning" In:
Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020), Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 1

Contextualising Data Mining within Educational Experiences

Keynote

Abelardo Pardo
University of South Australia

Abstract

The use of technology to mediate learning experiences provides an unprecedented amount of information that can be used to increase our understanding and improve the overall quality of those experiences. However, learning in general is strongly mediated by a very rich set of contextual factors. The two crucial steps to translate data into knowledge, sensemaking and deriving actions, are especially sensitive to these factors, and as such, need to be carefully considered to maximise positive outcomes. Areas such as personalisation are highly sensitive to the context in which each learner is engaged in an experience. Data-intensive techniques need to factor in these elements and assure learners are not adversely affected by situations ignored or inadequately handled by algorithms. This talk aims to explore how data mining applications can be properly situated to have a positive impact in specific aspects such as learning outcomes or connecting insights derived from data analysis with actions.

Biography

Abelardo Pardo is Professor and Dean of Programs (Engineering), at UniSA STEM, University of South Australia. His research interests include the design and deployment of technology to increase the understanding and improve digital learning experiences. More specifically, his work examines the areas of learning analytics, personalized active learning, and technology for student support.

He is the author of over 150 research papers in scholarly journals and international conferences in the area of educational technology and engineering education. He is currently member of the executive board and president of the Society for Learning Analytics Research (SoLAR).

Abelardo Pardo "Contextualising Data Mining within Educational Experiences" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 2

An AI-enabled Ecosystem for Learning & Assessment

Keynote

Alina von Davier
ACTNext

Abstract

AI-based tools, integrative technology and standards for various purposes in education have undergone significant development in the past few years. The vision is to build towards processes and/or parts thereof that are automatic and seamlessly integrated. In this presentation I will illustrate the architecture of a fluid infrastructure to effectively support learning and assessment systems. Each component is designed within a computational framework (AI blended with psychometrics) and each connection relies on construct taxonomy, database alignment, data exchange standards, and APIs.

I will describe a key AI-based content generator, Sphinx, developed at ACTNext. I'll use the ACTNext Educational Companion App as an example of how the pieces come together. Last but not least, I'll show how voice-based interface can be integrated within the versatile systems. The work has been conducted with an interdisciplinary team at ACTNEXT.

Biography

Alina von Davier, PhD., is the Chief Officer at ACTNext, a multidisciplinary innovation unit that is part of ACT and was founded in 2016. Her team is comprised of experts in fields ranging from psychometrics and learning sciences to software development, and artificial intelligence (AI) & machine learning (ML). Von Davier and her team operate at the forefront of Computational Psychometrics, an emerging interdisciplinary field concerned with the application of theoretical and data-driven computational methods and statistical modeling of multimodal, large scale/high dimensional learning and assessment data. Prior to leading ACTNext, von Davier was a senior research director at Educational Testing Service (ETS) where she led the Computational Psychometrics Research Center. Previously, she led the Center for Psychometrics for International Tests, where she was responsible for both the psychometrics in support of international tests, TOEFL[®] and TOEIC[®], and the scores reported to millions of test takers annually.

Von Davier is currently an adjunct professor at Fordham University and the president of the International Association of Computerized Adaptive Testing (IACAT). She currently serves on the board of directors for the Association of Test Publishers (ATP), and she is also a member of the board of directors for Smart Sparrow and of the advisory board for Duolingo.

Alina von Davier "An AI-enabled Ecosystem for Learning and Assessment" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 3

When is Deep Learning the Best Approach to Knowledge Tracing?

Theophile Gervet, Ken Koedinger, Jeff Schneider and Tom Mitchell
Carnegie Mellon University

Abstract

Intelligent tutoring systems (ITSs) teach skills using learning-by-doing principles and provide learners with individualized feedback and materials adapted to their level of understanding. Given a learner's history of past interactions with an ITS, a learner performance model estimates the current state of a learner's knowledge and predicts her future performance. The advent of increasingly large scale datasets has turned deep learning models for learner performance prediction into competitive alternatives to classical Markov process and logistic regression models. In an extensive empirical comparison on nine real-world datasets, we ask which approach makes the most accurate predictions, in what conditions. Logistic regression – with the right set of features – leads on datasets of moderate size or containing or containing a very large number of interactions per student, whereas Deep Knowledge Tracing leads on datasets of large size or where precise temporal information matters most. Markov process methods, like Bayesian Knowledge Tracing, lag behind other approaches. We follow this analysis with ablation studies to determine what components of leading algorithms explain their performance and a discussion of model calibration (reliability), which is crucial for downstream applications of learner performance prediction models.

Citation

Theophile Gervet, Ken Koedinger, Jeff Schneider and Tom Mitchell (2020). When is Deep Learning the Best Approach to Knowledge Tracing?. JEDM, Journal of Educational Data Mining, 12(3), (to be published).

Theophile Gervet, Ken Koedinger, Jeff Schneider and Tom Mitchell "When is Deep Learning the Best Approach to Knowledge Tracing?" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 4

Who's learning? Using demographics in EDM research

Luc Paquette
University of Illinois at
Urbana-Champaign
lpaq@illinois.edu

Alexandra Andres
University of Pennsylvania
alexandraandres@gmail.com

Jaclyn Ocumpaugh
University of Pennsylvania
ojaclyn@upenn.edu

Ryan Baker
University of Pennsylvania
ojaclyn@upenn.edu

Ziyue Li
University of Illinois at
Urbana-Champaign
ziyueli3@illinois.edu

Abstract

The growing use of machine learning for the data-driven study of social issues and the implementation of data-driven decision processes has required researchers to re-examine the often implicit assumption that data-driven models are neutral and free of biases. The careful examination of machine-learned models has identified examples of how existing biases can inadvertently be perpetuated in field such as criminal justice – where failing to account for racial prejudices in the prediction of recidivism can perpetuate or exasperate them – and natural language processing – where algorithms trained on human languages corpora have been shown to reproduce strong biases in gendered descriptions. These examples highlight the importance of thinking about how biases might impact the study of educational data and how data-driven models used in educational context may perpetuate inequalities. To understand this question, we ask whether and how demographic information, including age, educational-level, gender, race/ethnicity, socio-economic status (SES) and geographical location, is used in Educational Data Mining (EDM) research. Specifically, we conduct a systematic survey of the last five years of EDM publications that investigates whether and how demographic information about the students is reported in EDM research and how this information is used to 1) investigate issues related to demographics, 2) use the information as input features for data-driven analyses or 3) to test and validate models. This survey shows that, although a majority of publication reported at least one category of demographic information, the frequency of reporting for different categories of demographic information is very uneven (ranging from 5% to 59%) and only 15% of publications used demographic information in their analyses.

Citation

Paquette, L., Ocumpaugh, J., Baker, R., and Li, Ziyue (2020). Who's learning? Using demographics in EDM research. *JEDM, Journal of Educational Data Mining*, 12(3), (to be published).

Luc Paquette, Alexandra Andres, Jaclyn Ocumpaugh, Ryan Baker and Ziyue Li "Who's learning? Using demographics in EDM research" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 5

Analyzing Student Strategies In Blended Courses Using Clickstream Data

Nil-Jana Akpinar
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
nakpinar@stat.cmu.edu

Aaditya Ramdas
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
aramdas@stat.cmu.edu

Umut Acar
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
umut@cs.cmu.edu

ABSTRACT

Educational software data promises unique insights into students' study behaviors and drivers of success. While much work has been dedicated to performance prediction in massive open online courses, it is unclear if the same methods can be applied to blended courses and a deeper understanding of student strategies is often missing. We use pattern mining and models borrowed from Natural Language Processing (NLP) to understand student interactions and extract frequent strategies from a blended college course. Fine-grained clickstream data is collected through Diderot, a non-commercial educational support system that spans a wide range of functionalities. We find that interaction patterns differ considerably based on the assessment type students are preparing for, and many of the extracted features can be used for reliable performance prediction. Our results suggest that the proposed hybrid NLP methods can provide valuable insights even in the low-data setting of blended courses given enough data granularity.

Keywords

Student Strategies, Blended Courses, hybrid NLP methods

1. INTRODUCTION

Data collected through educational software systems can provide promising starting points to address hard questions rooted in the learning sciences. Modern education relies increasingly on these systems to assist teaching and grading, manage learning content, provide discussion boards, facilitate group work, or replace the traditional class room setting altogether. While blended courses revolve around the traditional class room setting accompanied by task-specific software support, Massive Open Online Courses (MOOCs) are usually entirely virtual and often involve video lectures and hundreds to thousands of students in a single course. Almost by design, these systems come with unprecedented opportunities for large scale data collection on students' study habits, content exposure and learning trajectories.

Much of the previous research effort has been directed towards performance prediction with the overall rationale that reliable estimation of students' grades and dropout probability at early course stages can be used to devise Early Warning Systems (EWSs) [e.g. 31, 19, 30, 7]. Despite considerable success in this area, many performance prediction models suffer from a list of shortcomings. Prior work on performance prediction from student online activity data has predominantly focused on MOOCs [e.g. 8, 28, 25], and it is unclear if the same methods can be applied to blended courses [3]. In most blended courses, some of the learning activity takes place offline and cannot be tracked which leads to relatively shallow data on only fragments of courses. In addition, many of the features that can be derived are simple and coarse summary statistics of students' online activity data, e.g. counts of clicks or logins, that only have a limited capacity to reflect the often complex strategies students take when interacting with course material.

A detailed understanding of how students interact with educational systems and the strategies they take is crucial for reliable performance prediction. We thus seek to understand how students approach learning in blended courses based on the second half of a sophomore level college course in computer science. Our data is drawn from Diderot, a non-commercial educational software system developed at Carnegie Mellon University which spans functions for virtually all course components outside of face-to-face class and recitation times, and thereby allows us to overcome many of the challenges that are generally faced when mining blended courses. Despite evident similarities, there are several important characteristics which differentiate our blended learning setting from the study of MOOCs. Most importantly, our data spans relatively few students and student actions which constitutes a challenge for many of the previously proposed methods. In addition, we have access to data that is unique to in-person classes such as individual attendance, and the nature of our activity data facilitates contextualization of student behavior which promises to increase the interpretability of downstream prediction models.

In this paper, we place a dual focus on methodology and educational insights. On the one hand, we propose new modeling pipelines based on ideas from natural language processing that work well in the low-data setting of blended courses. On the other hand, we apply both new and existing methods to Diderot data and gain valuable insights into student behavior while addressing the following research questions:

Nil-Jana Akpinar, Aaditya Ramdas and Umut Acar "Analyzing Student Strategies In Blended Courses Using Clickstream Data" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 6 - 17

- RQ1** How do students interact with course material, and what are frequent strategies they take?
- RQ2** How do students use these strategies for homework solving as compared to exam preparation?
- RQ3** Are student strategies indicative of grade outcomes?

The remainder of this paper is outlined as follows. We discuss related work in Section 2, and proceed to give some context for the data in Section 3. Section 4 describes our methods including the preprocessing of clickstream data, and we discuss our results in Section 5. Finally, conclusions are drawn in Section 6.

2. RELATED WORK

2.1 Analysis Of Online Student Behavior

Raw data from educational software systems often comes in the form of time-stamped student actions with an array of suitable identifiers. Evidence for correlations between activity log-based features and performance outcomes are plentiful. Many of the commonly discussed features revolve around simple summary statistics such as counts of certain types of actions, and have been shown to be indicative of students' success particularly in MOOCs. Recent lines of research find links between general course completion in MOOCs and the number of watched videos [39, 9], the number of question answer attempts [9], and the time spent on assignments [4]. Similar results have been observed for blended courses but are much scarcer [40, 16]. In [16], the authors analyze sequences of transitions between different online platforms in two undergraduate level college courses. Their study finds that, although students are generally more likely to stay on the same platform in a study session, high achieving students transition more often and are more likely to use the discussion board. In many cases, the limited amount of data in blended courses is problematic and can lead to complications such as zero-inflated count variables.

A major shortcoming of count-based methods is their failure to leverage the sequential structure of students' interactions with educational software systems. Both the order and the time difference between actions promise to carry valuable information that can be taken into consideration when relying on sequence based methods instead. In this work, we propose a pipeline for analyzing student online behavior based on session study sequences. While the order of actions is taken into account explicitly, time differences help us to derive reliable study sessions.

2.2 Study Sessions

Sequence-based approaches to processing online student activity data group student actions into smaller sessions. In the case of click actions, these sequences are generally referred to as clickstreams. The goal when breaking a flow of actions into session clickstreams is to maintain some notion of interpretability, i.e. to devise meaningful study sessions. While this appears to be easy in some cases, it is generally non-trivial to find automated cut-offs rules that find sensible representations of study sessions for a large and diverse set of clickstreams at once.

Previous research suggests several different strategies to split clickstreams. The authors of [8] choose fixed duration time frames to group student actions from a several months long MOOC. The researchers decide for durations between one day and one month and show some success in the downstream prediction of student achievements with their choices. Similar fixed durations are used in [2]. Another popular splitting strategy is based on time-out thresholds where a new sub-session is started when no action was performed in a predefined time window [32, 5, 36, 12, 13]. The authors of other studies go one step further and combine the approaches by first, splitting at a fixed duration cut-off and second, at data-driven timeout thresholds of 15 minutes for 'study sessions' and 40 minutes for 'browser sessions' [16]. Similar data-driven approaches are pursued in [45, 40]. Other common heuristics include splitting at navigational criteria such as reloading of the course page [26].

On a high level, the problem of devising meaningful sub-sessions is closely related to the problem of time-at-task estimation in web-usage mining. Ideally, study sessions reflect time periods in which students interact with the material without any major breaks or distractions. There is a rich body of literature on time at task estimation that suggests that there is no one-fits-all solution to finding suitable time windows to split activity streams at [e.g. 26, 6, 11]. Previous research suggests that the exact splitting heuristic can have a significant effect on overall model fit, model significance, and even interpretation of findings in the downstream modeling tasks [26]. In [26], the authors explore the effect of 15 different time-at-task estimation procedures on five different models of student performance. Overall, the authors conclude that there is no universally best method and recommend a mixture of existing methods including data-driven components. Following this suggestion, we employ a multi-step splitting procedure including navigational criteria, data-driven time-out thresholds, and separation of assessment weeks inspired by the procedure in [16].

2.3 Sequence Analysis

Different methods have been proposed to process sequence-type student action data dependent on the amount of data, the length of sequences, and the goal at hand. Several lines of research rely on Markov chains and hidden Markov models which lend themselves well to visualization of sequences, but can make quantification of group differences in outcomes challenging [15, 14, 20]. Another commonly used class of methods is clustering of activity sequences [13, 23, 17]. Using data from three large MOOCs, the authors of [23] draw on simple k -means clustering of sequences of interactions with video lectures and assessments and observe four high-level student trajectories: completing assessments, auditing the course, disengaging after a while, and sampling content. In order to cluster the sequences, the authors rely on a numerical translation of student actions. The authors of [13] cluster and visualize students' interactions with a college math environment, and instead rely on Levenshtein distance to measure the distance between sequences. Some works combine Markov models and clustering to account for the randomness introduced by the Markov models and report more robust results [41, 27, 24].

Although the described methods allow for a relatively easy

grouping of sequences, interpretation of clusters can be non-obvious. One way to address this problem is to deliberately focus on finding relevant sub-parts of action sequences. Methods based on this goal can be summarized under the term pattern mining, and are both wide-spread and diverse. A relatively recent approach is given by differential pattern mining which focuses on automatically extracting patterns that are both above a certain threshold in frequency, and sufficiently different among groups of interest (e.g. high and low achieving students) [22, 21]. Other lines of research rely on more traditional data mining techniques [18, 35], or extraction of n -grams, i.e. sub-sequences of n consecutive actions [8, 33, 44, 37]. The authors of [33] use a multi-step procedure to extract frequent n -grams that are subsequently used to identify different strategies in a collaborative interactive tabletop game. Part of our analysis is based on a similar approach to extract frequent behavioral patterns, and combines ideas of n -gram extraction and clustering to get more robust results.

A different class of promising methods is rooted in Natural Language Processing (NLP). Hybrid language models lend themselves well to the sequential structure of education data, and their use for student activity sequences has lead to some success in retrieving patterns and creating new visualizations. The underlying idea is that, given sufficiently fine-grained data, students' sequential actions resemble words building sentences and can be attributed some 'semantic meaning'. The NLP toolbox has not yet been explored fully, but some attempts to using language models for educational data are noteworthy and relevant for the context of our work. The authors of [44] use topical n -gram models to automatically extract 'topics' in the form of frequent patterns from clickstreams. In [37], the authors train a skip-gram neural network to receive a structure preserving vector embedding of the types of clicks student can make. After standard dimensionality reduction, the researchers are able to provide a new kind of visualization of students' trajectories through the course. Since modern NLP models generally require large amounts of granular training data, work relying on these models has exclusively focused on MOOCs so far. In this study, we draw on Latent Dirichlet Allocation (LDA) in order to automatically extract frequent patterns and compare derived student strategies against the results of a more traditional n -gram pipeline. In some sense, LDA is similar to the ideas proposed by [44] but requires less training data which renders it particularly useful for blended courses. In addition, we use an adapted form of the skip-gram model proposed by [37] in order to explore the context of student actions in our data. To the best of our knowledge, this is one of the first works to employ NLP methods for analysis of blended courses.

3. DATA

3.1 Data Context: Diderot

The data this study builds on was collected through the educational software system Diderot. Diderot is a cloud-based course support system commonly used to assist undergraduate and graduate level college courses. The system spans a wide range of functionalities including sharing of lecture notes, a discussion board (called post office), in-class attendance polls, homework submission, and automatic code grading. This bandwidth usually renders the use of addi-

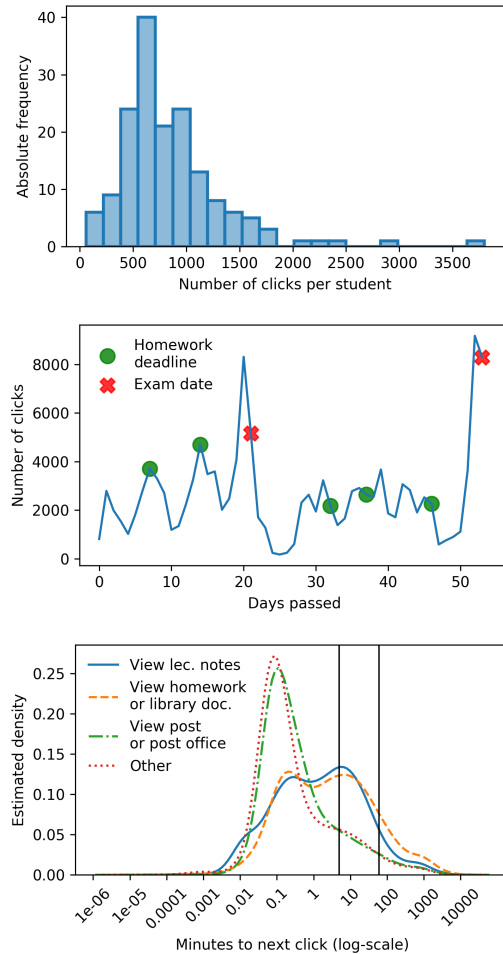


Figure 1: Histogram of the number of clicks per student. We observe 138,960 clicks spread between 164 students (top). Number of clicks over observation period with assessment deadlines highlighted (middle). Kernel density estimates for log-distribution of waiting times between clicks dependent on type of last click after splitting at assessment weeks and Load course actions. Final cut-offs at 5 and 60 minutes are indicated by vertical lines (bottom).

tional outside technological course support unnecessary. In turn, the student usage data collected from Diderot can give an almost comprehensive view on students' course participation outside of face-to-face class times.

When it comes to sharing of lecture notes, Diderot takes a more granular and interactive approach as compared to traditional learning management systems. Content is split into small sub-entities (called atoms) which are displayed in a linear fashion following the outline of a chapter. Atoms are highly interactive and come with a variety of clickable icons that allow students to take notes, bookmark, follow, or like atoms, and, in particular, to ask questions concerning their content. Discussions about course material that are sparked in this way are visually attached to the respective atom, allowing other students to submit comments. This setup results in much richer data on interactions with lecture notes

than we can expect from PDF formatted lecture material.

Most student usage data from Diderot is presented by individual, time-stamped click actions that come with various identifiers coding the exact type, user, and location of the interaction. In turn, the activity data can be broadly separated into navigation (e.g. `Load course`, `Click link`, `Search`), discussion (e.g. `Go to post office`, `Create post`), and behaviors (e.g. `Like atom`, `Follow post`).

3.2 Data Description And Exploration

Our data is drawn from the second half of a large sophomore-level computer science course taught at Carnegie Mellon University in spring 2019. Since data is not available for the first part of the course due to initial technical difficulties, we exclude all students who dropped the course throughout the semester. One additional student was excluded based on inflated click patterns which suggested an attempt at automatically scraping content. Along with the click data, we rely on performance information measured by homework and exam grades, as well as student-level lecture and recitation attendance logs. All data is collected through Diderot and matched based on anonymous student identifiers. A summary of the click data over the seven week observation period is displayed in Figure 1.

Types of clicks. At finest granularity, Diderot allows for several tens of thousands distinct click actions within a single course since every individual click is associated to a fully specified object and activity. However for the sake of analysis, we group clicks into different types where the appropriate level of granularity is non-obvious. We aggregate clicks based on the type of object they refer to as well as the activity performed. In order to maintain interpretability, this aggregation is performed separately in each sub-part of the course given by lecture notes, homework material, recitation notes, a library documentation (which is comprised of coding references), and practice exams. This leaves us with 37 different click types, the most common of which are summarized in Table 1.

Grades and types of assessment weeks. Performance outcomes are measured by percentage grades in five homeworks and two exams (a midterm and final exam) that fall into the observation period. This naturally divides the data into seven assessment weeks with a deadline for a homework problem set or exam at the end of each period. Deadlines are approximately evenly spaced with only one extended homework period of 11 days after the midterm exam (which also spans over a four day spring holiday), followed by a shorter homework period of only 5 days. We take interest in relating students’ study behavior to two distinct outcome variables: (1) The type of the assessment week, i.e. homework deadline or exam, and (2) the percentage grade students received in the respective assessment. As depicted in Figure 1, there are visible spikes of increased activity before the assessment week deadlines especially before the two exams. In addition, we note that the distribution of grades appears notably different between homeworks and exams which is confirmed by a two-sample Kolmogorov-Smirnov test ($p < 0.001$). While the distribution of exam grades is approximately bell-shaped with heavy tails and a slight left-skew, i.e. more particularly high scores than particularly low scores, the homework grade

Table 1: Summary of the most frequent click types.

Click type	Count	Share
View chapter in lecture notes	24,420	17.57 %
View general post	21,555	15.51 %
Load course	19,677	14.16 %
View post office	16,231	11.68 %
View atom post	15,468	11.13 %
View homework atom	7,888	5.68 %

distribution is left-skewed with additional modes at 0 and 100. This difference in distributions is unsurprising as exams are generally graded on a curve and cannot be skipped by students, while homeworks allow for more variability.

Class attendance. Attendance in lecture and recitation sessions was taken with Diderot polls. If a student participated in the poll, which was generally only open for a few minutes, it was assumed that they attended the session. We treat attendance in lectures and recitations separately and aggregate the binary information on an assessment week basis by taking the mean. In turn, student’s attendance scores lie between 0 and 1 with the exception of the final exam week which is not associated to any contact class time.

4. METHODS

4.1 Session Clickstreams

In raw form, each student is associated with a single clickstream which consists of ordered click actions over the whole observed time period. We employ a multi-step procedure to split this data into more meaningful study sessions. First, we divide the clickstreams based on assessment weeks. Second, we split the resulting sub-clickstreams each time a `Load course` action is recorded, and last, we choose a data-driven timeout threshold to further break up the resulting sequences.

In order to find a suitable timeout threshold, we employ a technique similar to [16] and examine the distribution of time differences in the sub-sequences. We find that the distribution of waiting times supports a wide range but is rapidly decaying. While 75 % of clicks are made within 2.81 minutes or less, a small subset of clicks has time differences of up to 7 days. Figure 1 shows kernel density estimates of log-transformed minutes until the next click within the sub-clickstreams obtained after the second step of our procedure. Different estimates are obtained for distinct categories of actions. While the logarithmic distribution of post-related and miscellaneous clicks is unimodal with the majority of follow-up clicks made within one minute, the distribution for clicks related to homework and lecture notes has an additional mode at about 5-10 minutes. This disparity is unsurprising given that most actions can be expected to be short, while reading through lecture notes or homeworks can be a more lengthy process. In order to preserve both types of sessions, we separate clickstreams at a 60 minutes threshold if the last action was loading of lecture notes or homework related content, and at 5 minutes otherwise. As a result, we obtain a total of total of 35,703 session clickstreams where each clickstream has between one and 115 clicks with mean

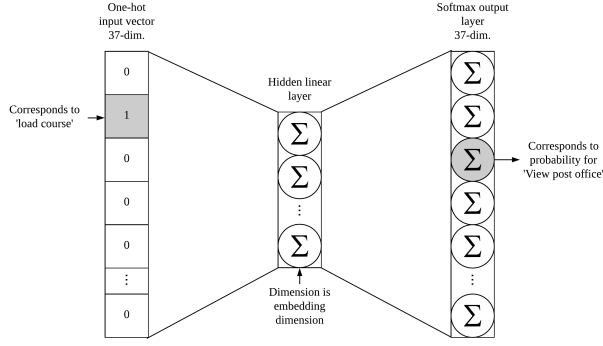


Figure 2: Skip-gram neural network. The hidden layer linearly transforms one-hot encoded inputs while the softmax output layer approximates the probability that each given click type appears in the same context as the input click. After training, the weights of the hidden layer provide a structure preserving embedding of click types.

of 3.98 clicks and standard deviation of 6.23; 75% of session clickstreams have at most 4 clicks.

4.2 Context Of Click Types

We explore the contexts in which different types of clicks are made in order to gain some understanding of how students generally use the course support system. This is crucial since Diderot is a fully integrated interactive platform that allows the same type of click in contexts that can have different interpretations. Inspired by [37], we tackle this problem by devising a structure-preserving embedding of the click types into a real-valued vector space, i.e. each click type is mapped onto a vector such that click types that appear in the same contexts or are interchangeable are close to each other. This type of embedding can be obtained from a skip-gram model which is a common supervised two layer neural network model often used for language type data (see Figure 2).

Training data for the model is build by extracting pairs of neighboring click types from the session clickstreams. More concretely, each input click is paired with each click appearing within some index in the same clickstream. Both the window size and the number of hidden units are important hyperparameters. Since most of our clicks are short and we seek an embedding of only 37 clicktypes, we explore small values for both parameters, i.e. window sizes in $\{1, 2\}$ and embedding sizes in $\{3, 4\}$. After this small grid search, we only retain the model with the lowest average training loss in the last 2000 training steps. In order to speed up training, we rely on mean noise-contrastive estimation (NCE) loss where 8 negative classes are sampled for every batch instead of computing the entire softmax output. All models are trained over a maximum of 300,000 training steps with SGD with learning rate 1 and a batch size of 512. Training is terminated early when the average loss over 2000 training steps does not change considerably for 5 consecutive non-overlapping 2000-step periods. Because training the model is only the surrogate task in order to obtain the embedding, we train on all available data which comprises 206,514 or 363,260 pairs dependent on the window size.

4.3 Frequent Pattern Extraction

4.3.1 Clustered n -grams

We refer to finite sub-sequences of clickstreams as frequent patterns if they appear various times across different students, study sessions, and assessment weeks. Our goal is to automatically extract frequent patterns which represent some kind of strategy or high level task students are fulfilling. As an example, the sequence [Login - View post office - View general post] could be interpreted as an attempt to catch up on the course news.

Pattern mining in educational data mining can lead to relatively unstable results. In order to increase robustness, we examine and compare the results of two distinct procedures for frequent pattern extraction. The first method resembles the procedure proposed by [33], and consists of a multi-step procedure which first extracts a large set of candidate patterns, and then narrows the selection down by similarity grouping. Formally, we proceed according to the following steps:

- (1) **All n -grams.** We extract n -grams, i.e. consecutive sub-sequences of n clicks, from the session clickstreams. Since we expect very short patterns to be uninterpretable, and particularly long patterns are rare in our dataset, we choose $n = 3, 4, 5$.
- (2) **Candidate patterns.** Only the most frequent patterns are kept as candidates for further analysis. Following some experimentation, we choose to keep the most frequent 1 % of patterns of each length.
- (3) **Hierarchical clustering.** The set of candidate patterns can be expected to be repetitive in the sense that patterns might be similar but vary in length or differ in a single click action but yield the same interpretation. To address this issue, we automatically group candidate patterns by agglomerative clustering with average linkage. The number of clusters, and thus of final frequent pattern categories, is chosen by visual inspection of the model's dendrogram.

The final step of this procedure requires us to specify a notion of similarity between patterns. In some sense, it is natural to draw on a string distance measure as sequences of clicks resemble many of the characteristics we would expect from natural language. While the authors of [33] draw on the traditional Levenshtein distance, we choose the Jaro-Winkler distance between two patterns p_1, p_2 measured by $1 - jw(p_1, p_2)$, where $jw(\cdot, \cdot)$ denotes the Jaro-Winkler similarity. Jaro-Winkler distance is an adaptation of more traditional edit distances which takes the sequence length as well as common starting sub-sequences into account. This allows more sensible measuring of similarities between repetitive patterns of different lengths such as the 3-gram [View general post - View general post - View general post] and the 5-gram [View general post - View general post - View general post - View general post - View general post]. Intuitively, the two patterns should have a low distance and in fact, their Jaro-Winkler distance is approximately 0.093 while their normalized Levenshtein distance is 0.4. For our

purpose, we treat each click as a character that can be exchanged or transposed for a penalty on the distance. Then, the Jaro similarity is defined as

$$j(p_1, p_2) := \begin{cases} 0 & \text{if } m = 0, \\ \frac{1}{3} \left(\frac{m}{|p_1|} + \frac{m}{|p_2|} + \frac{m-t}{m} \right) & \text{else,} \end{cases}$$

where m is the number of matching clicks within an index window of $\lfloor (\max\{|p_1|, |p_2|\}/2) \rfloor - 1$, and t is half the number of required transpositions for matching clicks. Further, the Jaro-Winkler similarity is defined as

$$jw(p_1, p_2) := j(p_1, p_2) + \frac{l}{10}(1 - j(p_1, p_2)),$$

where l is the length of a common starting sequence between p_1 and p_2 (at most 4). The additional scaling ensures that distances are normalized to lie in $[0, 1]$.

4.3.2 Topic Model

The clustered n -grams procedure of extracting frequent patterns is easy to implement and model-free. However, it requires us to choose several hyperparameters such as the size of n -grams, the share of candidate patterns, or the number of clusters. It is also likely that the exact choice of the edit distance in the clustering step has a non-negligible effect on the observed results. In order to test our results for robustness, we employ a second method for pattern extraction and compare the resulting student strategies. This method draws on the idea that session clickstreams resemble sentences, individual clicks resemble words, and there is some notion of semantic to a sequence of clicks. Based on these similarities, we use Latent Dirichlet Allocation (LDA), a common NLP model that allows automatic extraction of topics from written documents.

LDA is a Bayesian model which, in our case, is build on the assumption that each session clickstream is a mixture of patterns and each pattern is a mixture of clicktypes. We use the words pattern and topic interchangeably here. While the clickstreams (and hence clicktypes) are given to the model, the topics are latent and can be inferred from the fitted model. The prior on the session clickstream generation assumes that M clickstreams of lengths N_1, \dots, N_M are drawn according to the following steps. (1) Draw a topic distribution $\theta_i \sim \text{Dir}_k(\alpha)$ for each $i = 1, \dots, M$, where k is the number of topics. (2) Draw a click type distribution for topics $\phi_i \sim \text{Dir}_V(\beta)$ for each $i = 1, \dots, V$, where V is the number of different click types. (3) For each click position i, j with $i \in \{1, \dots, M\}$ and $j \in \{1, \dots, N_i\}$, first, choose a topic according to $z_{ij} \sim \text{Multinomial}(\theta_i)$, and second, draw a click type from $w_{ij} \sim \text{Multinomial}(\phi_{z_{ij}})$. LDA comes with three hyperparameters: the prior Dirichlet parameters α and β which express some prior belief on how the mixtures of topics and click types are composed, and the number of latent topics k . While we set the prior Dirichlet parameters to suggested default values, i.e. normalized asymmetric priors, the number of latent topics requires some more thought. Recent research suggests the use of topic coherence measures for comparison of models with different choices of k [34, 43]. On a high level, topic coherence attempts to measure semantic similarity between high scoring words (or here click types) in each topic which gives some indication of how interpretable the topics in question are. We experiment with

several numbers of topics ranging around the number of frequent patterns extracted by the clustered n -gram technique. Since no significant differences in coherence can be observed, we resort to using the same number of topics as for the clustered n -gram method for the sake of comparison.

4.4 Prediction Models

Frequent patterns counts as features. In order to explore what role the extracted strategies play in homework solving versus exam preparation and whether they drive success, we build two prediction models based on patterns counts from the clustered n -gram method. For this, a representative pattern of 3 clicks is chosen for each of the devised strategy clusters, and its occurrences in each of the session clickstreams is counted by comparing against each 3-gram derived from the clickstream. Since we cannot expect the chosen pattern to accurately represent the whole cluster, we allow a Jaro-Winkler distance up to 0.2 when comparing the sub-sequences. This procedure allows matching of click sequences with only one replacement ($1 - jw(abc, abd) \approx 0.18$), one transposition ($1 - jw(abc, acb) \approx 0.10$), or one replacement and one transposition ($1 - jw(abc, adb) \approx 0.20$). In order to build student and assessment week based prediction models, we aggregate pattern counts along assessment weeks and individual students by simple addition. Similar methods have been employed by [8, 29, 42, 10].

Predicting assessment type. A random forest classifier is trained to predict the assessment type, i.e. homework or exam, from frequent pattern counts, the number of clicks, and the number of session clickstreams a student has within a given week. In practice, it is unlikely that we would need to predict the assessment type as it is usually known. However when paired with careful analysis of feature importance and partial dependence, such model can yield valuable insights into the most important differences in student behavior between homework and exam weeks. We use 80 % of the 1,148 student-week combinations for training and hold back 20 % as test set. Hyperparameters including the maximum tree depth, the maximum number of features to consider at splits, the minimum number of samples per leaf, and the number of trees are chosen by a grid search over a range of values, where models are trained with 5-fold cross validation on the training set. Our model draws on Gini impurity to measure the quality of splits, and we evaluate feature importance based on the mean decrease in impurity (MDI) associated with splitting at a given feature when predicting Y . For a set of fitted trees $\mathcal{T} = \{T_1, \dots, T_N\}$, the MDI of a feature X_m is defined as

$$\text{MDI}(X_m) = \frac{1}{N} \sum_{T \in \mathcal{T}} \sum_{t \in T: v(s_t) = X_m} p(t) \Delta i(s_t, t), \quad (1)$$

where $p(t)$ is the proportion of samples that reaches node t , $v(s_t)$ is the variable used to split s_t , and $\Delta i(s_t, t)$ is the decrease of impurity generated by the split.

Predicting grade outcomes. Similar to the assessment type prediction model, we train a random forest regressor to predict students' grade outcomes based on strategy counts, the number of clicks, the number of session clickstreams, and attendance information. The additional consideration of lecture and recitation attendance requires us to remove all observations from finals week, since no face-to-face class time

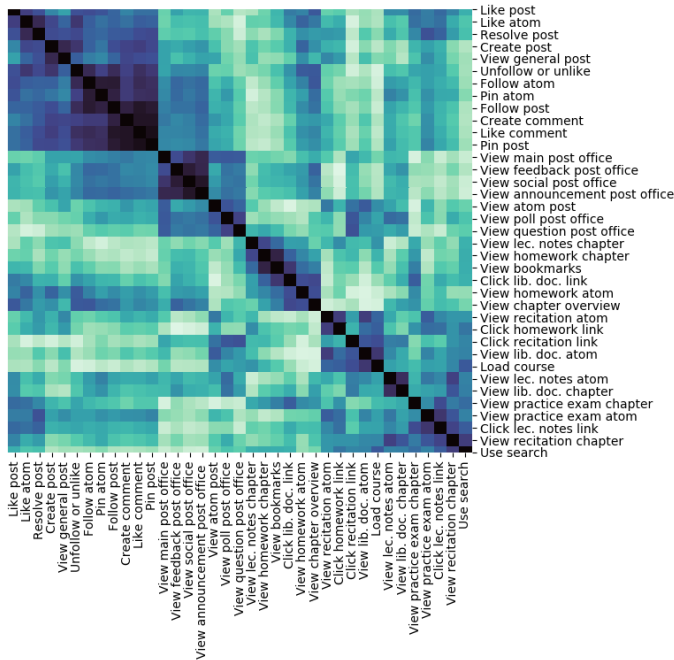


Figure 3: Euclidean distances of click type embeddings based on skip-gram neural network. Darker color suggests that embedding are close. Proximity in the embedding space suggests clicks generally appear in similar contexts. Rows and columns are clustered for visualization.

has taken place in the last week of the course. Since this constitutes half of all exam observations in the data and grade distributions of homeworks and exams are significantly different ($p < 0.001$), we limit our prediction model to homework grade prediction entirely. Of the 820 homework samples, 80 % are used for training and 20 % for testing. A grid search of hyperparameters with 5-fold cross-validation on the training set is performed, and feature importance is measured analogous to Equation 1 with the MSE as impurity measure.

5. RESULTS

5.1 RQ1 How do students interact with course material, and what are frequent strategies they take?

5.1.1 Context Of Click Types

In order to gain some initial understanding of online student behavior, we explore the contexts in which different types of actions are performed by deriving a skip-gram neural network based embedding of actions. After exploring a small grid of hyperparameter values, our skip-gram is trained on data pairs with window size 1 to learn a 4-dimensional embedding. Figure 3 depicts the Euclidean distances between the embedding vectors of different click types based on the model. Proximity of embeddings suggests that click types either appear in a similar context, i.e. within a few clicks of each other, or are interchangeable actions, i.e. have the same context. In other words, by exploring which actions lie close to a given click type in the embedding space, we can gain some insight into the set of clicks students typically

make right before and after. It is noteworthy that some types of actions appear together by design of the Diderot system, e.g. in order to comment on a post, the post has to be loaded. Figure 3 reflects many of these expected relations which gives some validation to our methodological approach.

Our results suggest several broad clusters of student actions. The block in the upper left corner of Figure 3 appears to focus on active discussion participation including click types such as **Like post** or **Create comment**. The next block is somewhat close to many of the active discussion actions and concentrates on scrolling through the discussion board represented by **View post office** type actions. Although more rigorous statistical analysis is needed, the results suggest some interesting interpretations:

- (1) **Students ask more questions about homeworks than about any other course materials.** This interpretation is based on the proximity of **Create post** to **View homework atom** which appears to be much closer than any other **View atom** type action. This suggests that student questions, comments and clarifications are more common for homework material than for lectures notes, recitation material, practice exams, or the library documentation.
- (2) **Students are more likely to interact with course-wide posts than material specific discussions.** The action **View general post** is close to interactive behavior such as **Create comment**, **Like post** or **Follow post** while **View atom post** appears to be performed mostly in a different context. This suggests that discussion-specific reactions and interactions concentrate mostly on general posts such as course announcements or social posts and are less common for questions and comments concerning particular parts of the course materials.

Overall, context analysis for click types based on skip-gram neural networks provides us with some valuable understanding of students' use of Diderot. The same method might be useful to other practitioners, in particular, for initial exploration of data collected through educational software systems. It appears that interpretable low-dimensional embeddings of a medium number of action types can be obtained with only a few weeks worth of data from a single college course which renders this method particularly useful for blended courses.

5.1.2 Frequent Pattern Extraction

Patterns are extracted with two distinct methods, and subsequently interpreted in terms of underlying student strategies. A summary of the results and comparison between the methods is given in Table 2. The left side of the table shows the results of the clustered n -gram pipeline for pattern extraction. The most frequent 1% n -grams for each $n = 3, 4, 5$ are extracted from the session clickstreams. This yields a candidate set of 223 sequential patterns which are clustered into 9 groups based on agglomerative clustering with average linkage and Jaro-Winkler distance as distance function. The number of clusters is informed by visual inspection of

Table 2: Comparison of student strategies extracted by clustered n -gram method and LDA. Patterns in the first block (B1) consist of exactly the same click types, while other patterns show differences but allow for similar interpretations (B2). Lastly, the LDA method finds a mixture of practice exam related patterns and a new load course pattern (B3).

	Clustered n -gram method		LDA method	
	Student strategy	Associated click types	Student strategy	High weight click types
B1	Look at lecture notes	View lecture notes chapter (75.88 %)	Look at lecture notes	View lecture notes chapter (1)
	Look at homeworks	View homework chapter (100 %)	Look at homeworks	View homework chapter (0.826)
	Look at recitation material	View recitation chapter (100 %)	Look at recitation material	View recitation chapter (0.712)
B2	Catch up on news	View general post (52.04 %), View main post office (24.3 %), View atom post (16.13 %)	Catch up on news	View general post (0.543), View main post office (0.410)
	Active homework engagement	View atom post (50 %), View homework atom (31.02 %), View general post (10.65 %)	Active homework engagement	View atom post (0.653), View homework atom (0.344)
	In-depth review of lecture notes	View lecture notes atom (50 %), View atom post (28.57 %), View lecture notes chapter (21.43 %)	In-depth review of lecture notes	View lecture notes atom (0.483), View atom post (0.31), Click link lecture notes (0.195)
	Look at library documentation	View library documentation chapter (85.29 %)	Look at library documentation	View library documentation chapter (0.674), Search atom (0.321)
B3	Go through a practice exam	View practice exam atom (100 %)	Practice exams	View practice exams chapter (0.658), View practice exams atom (0.341)
	Look at practice exams	View practice exams chapter (100 %)	Load course	Load course (0.998)

the respective dendrogram. It is noteworthy that the clusters appear to have imbalanced sizes with the largest cluster including 106 candidate patterns, and the smallest clusters containing only 2 or 3 of the candidate patterns. Yet, inspection of the associated click types and their in-cluster frequencies allows for intuitive interpretations as student strategies. Multiple of the devised strategies revolve around passive review of materials such as lecture notes, homeworks, recitation material, library documentation (which includes code snippets for reference), or practice exams. More involved strategies are given by active homework engagement, in-depth review of lecture notes, catching up on course news, and going through practice exams. For example, the catching up on course news strategy is associated with sequential patterns involving reading of general posts, atom posts, and loading the main post office page.

The right side of Table 2 summarizes the results of pattern extraction based on Latent Dirichlet Allocation (LDA). For the sake of comparison, we keep the number of extracted patterns fixed and derive 9 student strategies. By assumption of the model, each pattern is a mixture of all click types. In turn, extraction of weights is straightforward and we report the click types with highest weights for each pattern. We find that multiple of the extracted patterns match exactly the patterns retrieved with the clustered n -gram method in the sense that they are based on exactly the same click types (B1). Another set of patterns shows small changes in included click types, but essentially provides the same interpretation as the patterns found with the first method (B2). Lastly, the LDA method finds a practice exam strategy which broadly presents a mixture of the two practice exam related strategies from the first model, and a load course strategy which almost entirely consists of the **Load course** action (B3). The load course pattern likely arises from the session clickstreams with a single click

which present 30.30% of the session clickstreams. A total of 56.66% of these one-click sequences are **Load course** actions. Reasons for these single **Load course** clicks can be manifold. In some cases, students might get distracted immediately after loading the course, or they have to reload the course multiple times. However, we hypothesize that in most cases, the course overview page which is loaded when loading the course provided all information the student was looking for since it includes recent updates, posts and announcements. Contrary to the clustered n -gram method which only takes into consideration session clickstreams of at least three clicks, LDA can leverage even these short clickstreams. Yet, the additional insights gained through the load course pattern are marginal since it very short and hard to interpret as a strategy.

All in all, both methods roughly extract the same strategies which speaks in favor of the validity of both approaches. One could argue that the clustered n -gram method yields slightly more tangible insights since the patterns present actually frequently occurring sub-sequences. However for larger data sets, the method can become computational expensive rendering LDA a better choice.

5.2 RQ2 How do students use these strategies for homework solving as compared to exam preparation?

We extract strategy features for assessment week level prediction models by matching session clickstreams against the extracted frequent patterns. The results are summed up for each student-week combination and thus roughly represent how often a given student has used a strategy in a given assessment week. After this aggregation, 91.03 % show at least one occurrence of one of the patterns. We generally expect not all student click behavior to follow the extracted strate-

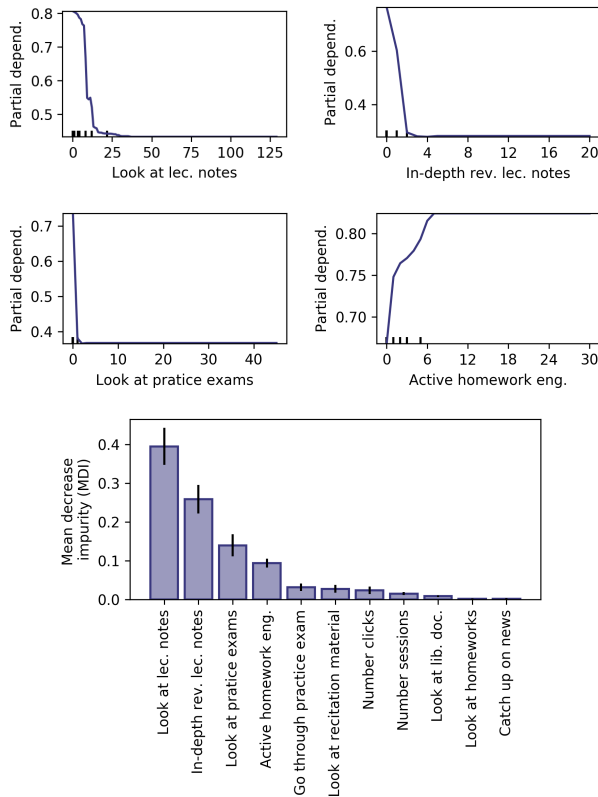


Figure 4: Relative feature importance for assessment week random forest prediction (1 = homework, 0 = exam) along with 95 % confidence intervals (bottom) and partial dependence plots for the most important features (top). Features include strategy counts from the clustered n -gram method, the number of clicks and the number of session clickstreams.

gies or stringent strategies at all. Thus, it is unsurprising that some of the student week combinations do not involve any of the patterns.

We train a random forest classifier to predict the assessment type on pattern counts, the number of clicks and the number of sessions within a given week. A total of 80 % of the data is used for hyperparameter tuning and training, while 20 % is withheld for testing. The model reaches a classification accuracy of 93.68 % on the training data which constitutes an evident improvement over the naive majority class prediction (71.90 % of the training data have the label homework). Based on a permutation test, we find that the model performs better than random on the training set ($p < 0.01$). A total of 100 permutations of labels were used for this evaluation. Accuracy on the test set is 93.91 % which suggests sufficient generalization ability of the prediction model.

The prediction model results suggest that students use the educational support system differently and employ the different strategies at different rates when preparing for exams as compared to doing homeworks. We examine feature importance in the model in order to gain more insights into these differences. Figure 4 depicts the mean decrease in impurity (MDI) for splits at the different covariates, as well as partial dependence of the predictions on the most impor-

tant features. We see that predictions are mainly driven by pattern counts of the strategies look at lecture notes (MDI = 0.395), in-depth review of lecture notes (MDI = 0.259), look at practice exams (MDI = 0.140), and active homework engagement (MDI = 0.094). Partial dependence plots show that while increased counts in the strategies related to lecture notes and practice exam engagement increase the probability that the model predicts an exam week, higher counts in the active homework engagement strategy increase the models likelihood of predicting an upcoming homework deadline. These results suggest that students approach to learning is driven by the kind of performance assessment they are given. It appears that the increased activity in exam weeks (see Figure 1) is largely based on increased engagement with lecture notes and practice exams, while interactions with the homework related content is generally less pronounced.

5.3 RQ3 Are student strategies indicative of grade outcomes?

We train a random forest regression model to predict homework grades on a individual week and student level. Features include students’ strategy counts, the number of clicks, the number of sessions, and the mean attendance in both lectures and recitations. Training is conducted on 80 % of available data while 20 % are withheld for testing. After hyperparameter tuning with 5-fold cross validation, the prediction model realizes a MSE of 0.046 on the training data set. A permutation test based on 100 permutations of labels shows a significant improvement over random performance with this model ($p < 0.01$). On the test set, the model attains a prediction MSE of 0.054 which suggests sufficient generalization ability.

Figure 5 explores the importance of the different features for predictions and displays partial dependence relations for the most important covariates. Since we use MSE as impurity measure, the mean decrease in impurity (MDI) for a given feature effectively corresponds to the mean decrease in variance we receive by splitting at the feature. We see that, in fact, the most relevant features appear to be the number of clickstream sessions (MDI = 0.311), the number of clicks (MDI = 0.221), lecture attendance (MDI = 0.123), and recitation attendance (MDI = 0.112). Partial dependence plots reveal that increases in any of the above features increase the predicted homework score percentage by a relatively large margin of up to 20 percentage points. Conversely, strategy counts appear to be less relevant for grade predictions with some exceptions. Most notably, the predicted grade rises with the number of times students actively engaged in homeworks (MDI = 0.077).

Overall, our results show some success in prediction of homework grade outcomes. The extracted features, including some of the pattern counts, add valuable information to the prediction model. In particular, students who come back to Diderot more often and thus use an increased number of study sessions to solve their homeworks, and students who generally interact with the system at high rates are predicted to have better grade outcomes. In addition to time at task, the mere attendance in lectures and recitations increases students’ grade outcome predictions. In fact, students in the our data set who attended at least one lecture in a given

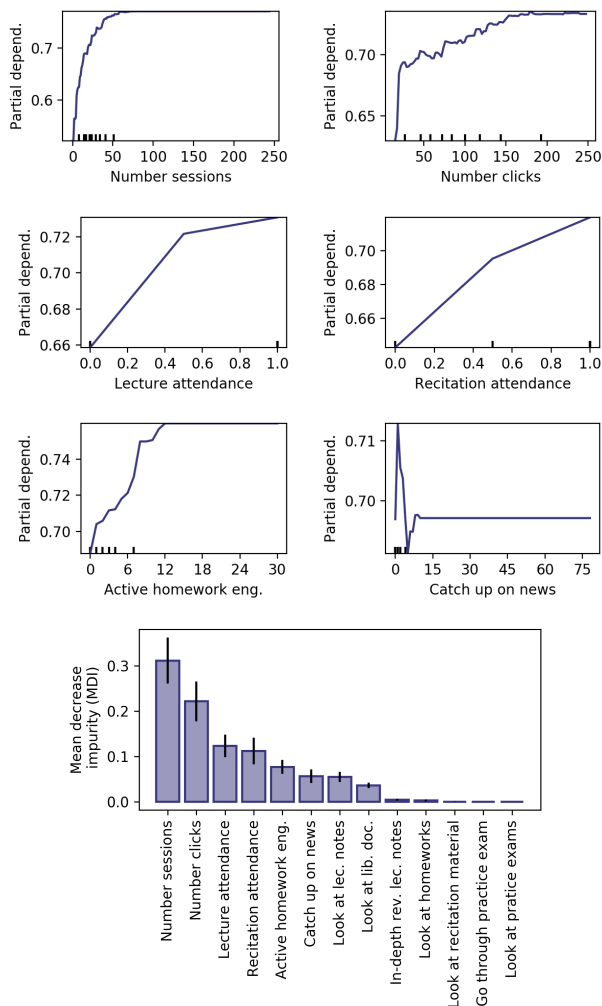


Figure 5: Relative feature importance for homework grade random forest prediction along with 95 % confidence intervals (bottom) and partial dependence plots for the most important features (top). Features include strategy counts from the clustered n -gram method, the number of clicks and session clickstreams, and attendance information.

assessment week on average received a homework percentage grade of 76.58 %, while students who skipped lectures on average scored 55.86 %. For recitation attendance this corresponds to 74.34 % and 49.20 % respectively.

Both of the discussed prediction models provide valuable insights for instructors and educational system design. The tree-based ensemble methods are particularly suitable for initial modeling and processing of features on different scales. Their main advantage over many other models is the relatively straightforward explainability of predictions given partial dependence plots and measures of feature importance which renders them a useful approach to high stakes at-risk prediction.

6. CONCLUSIONS

Data from educational software systems provides insights into students’ study behaviors. While performance predic-

tion in MOOCs has been explored extensively, similar studies for blended courses are scarce and often lack a deeper understanding of the underlying student strategies. Based on fine-grained contextualizable click data collected through the non-commercial course support system Diderot, we explore how students interact with educational software systems, which strategies they employ to engage with course materials and in which ways strategies depend on the assessment type and drive performance. Our contributions are two-fold: (1) We gain relevant understanding of students’ learning behavior that both confirms and adds to the existing literature. (2) We propose new NLP-inspired approaches to analyzing student strategies’ based on clickstream data in blended learning scenarios which typically come with moderately sized data sets.

On the educational side, our results provide valuable insights into how students interact with course systems. In line with previous research [38, 1], we observe increased activity before deadlines, and, in particular, in the days leading up to an exam. Exam preparation appears to come with increased review of lecture notes as compared to homework solving. In general, students seem to ask more questions related to homeworks as compared to other class materials such as lecture notes, recitation materials or practice exams. At the same time, interactions with already existing posts such as liking or commenting seems to concentrate mostly on course-wide announcements, social posts and course feedback discussions and appears to be less common for direct questions on course materials. Many of the derived features have some predictive power for performance outcomes. In particular, the number of study sessions, the number of clicks, attendance in lecture and recitation, and engaging with homework related course content are strong predictors for homework grades in our model. The described observations are entirely based on data from a seven week period of a large sophomore level college course since technical difficulties prohibited collection of data for the remainder of the semester. In the future, more complete data (e.g. from an entire course, or even multiple courses such as the same course offering over several years) could provide an enhanced understanding of student behavior and allow the tackling of more complex problems such as the simultaneous prediction of homework and exam grades which, such as in our data, can have very different distributions.

The methods proposed in this work promise to be useful to a broad range of researchers and practitioners who find themselves analyzing activity log-data from blended courses, or are at the initial stages of developing early warning systems. The key insight of this work is that hybrid NLP methods can be used to thoroughly analyze contexts of actions as well as frequent strategies in the relatively low-data setting of blended courses. To the best of our knowledge, similar models have previously only been employed in the setting of MOOCs [e.g. 44, 37]. In fact, our analysis shows that topic models such as latent Dirichlet allocation can recover almost the same student strategies as more traditional data mining based pipelines of pattern extraction, and small versions of skip-gram neural networks can provide valuable insights into the context of student actions even with moderately sized data sets.

References

- [1] L. Agnihotri, A. Aghababayan, and S. Mojarad. Mining Login Data For Actionable Student Insight. In *Proceedings of the 8th International Conference on Educational Data Mining*, pages 472–475, 2015.
- [2] B. Amnueypornsakul, S. Bhat, and P. Chinprutthiwong. Predicting Attrition Along the Way: The UIUC Model. In *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 55–59, 2014.
- [3] T.-S. An, C. Krauss, and A. Merceron. Can Typical Behaviors Identified in MOOCs be Discovered in Other Courses? In *Proceedings of the 10th International Conference on Educational Data Mining*, pages 220–227, 2017.
- [4] J. M. L. Andres, R. S. Baker, D. Gašević, G. Siemens, S. A. Crossley, and S. Joksimović. Studying MOOC completion at scale using the MOOC replication framework. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, pages 71–78, 2018.
- [5] H. Ba-Omar, I. Petrounias, and F. Anwar. A framework for using web usage mining to personalise e-learning. In *Proceedings of the 7th IEEE International Conference on Advanced Learning Technologies*, pages 937–938, 2007.
- [6] R. S. J. D. Baker. Modeling and Understanding Students’ Off-Task Behavior in Intelligent Tutoring Systems. In *Proceedings of ACM SIGCHI: Computer-Human Interaction*, 2007.
- [7] B. Bakhshinategh, O. R. Zaiane, S. ElAtia, and D. Iperciel. Educational data mining applications and tasks: A survey of the last 10 years. *Education and Information Technologies*, 23(1):537–553, Jan. 2018.
- [8] C. Brooks, C. Thompson, and S. Teasley. A time series interaction analysis method for building predictive models of learners using log data. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 126–135, 2015.
- [9] Y. Chen and M. Zhang. MOOC student dropout: pattern and prevention. In *Proceedings of the ACM Turing 50th Celebration Conference - China*, pages 1–6, 2017.
- [10] C. A. Coleman, D. T. Seaton, and I. Chuang. Probabilistic Use Cases: Discovering Behavioral Patterns for Predicting Certification. In *Proceedings of the Second (2015) ACM Conference on Learning at Scale*, page 141–148, 2015.
- [11] R. Cooley, B. Mobasher, and J. Srivastava. Data Preparation for Mining World Wide Web Browsing Patterns. *Knowledge and Information Systems*, 1(1):5–32, Feb. 1999.
- [12] R. del Valle and T. M. Duffy. Online learning: Learner characteristics and their approaches to managing learning. *Instructional Science*, 37(2):129–149, 2009.
- [13] M. C. Desmarais and F. Lemieux. Clustering and Visualizing Study State Sequences. In *Proceedings of the 6th International Conference on Educational Data Mining*, pages 224–227, 2013.
- [14] L. Faucon, L. Kidzinski, and P. Dillenbourg. Semi-Markov model for simulating MOOC students. In *Proceedings of the 9th conference on Educational Data Mining*, pages 358–363, 2016.
- [15] C. Geigle and C. X. Zhai. Modeling MOOC student behavior with two-layer hidden markov models. In *Proceedings of the 4th ACM Conference on Learning at Scale*, pages 205–208, 2017.
- [16] N. Gitinabard, S. Heckman, T. Barnes, and C. F. Lynch. What will you do next? A sequence analysis on the student transitions between online platforms in blended courses. *arXiv: 1905.00928*, 2019.
- [17] J. Guerra, S. Sahebi, P. Brusilovsky, and Y.-r. Lin. The Problem Solving Genome: Analyzing Sequential Patterns of Student Work with Parameterized Exercises. In *Proceedings of the 7th International Conference on Educational Data Mining*, pages 153–160, 2014.
- [18] J. Herold, A. Zundel, and T. F. Stahovich. Mining Meaningful Patterns from Students’ Handwritten Coursework. In *Proceedings of the 6th International Conference on Educational Data Mining*, pages 67–73, 2013.
- [19] Y.-H. Hu, C.-L. Lo, and S.-P. Shih. Developing early warning systems to predict students’ online learning performance. *Computers in Human Behavior*, 36:469–478, July 2014.
- [20] H. Jeong and G. Biswas. Mining Student Behavior Models in Learning-by-Teaching Environments. In *Proceedings of the 1st International Conference on Educational Data Mining*, pages 127–136, 2008.
- [21] J. S. Kinnebrew and G. Biswas. Identifying Learning Behaviors by Contextualizing Differential Sequence Mining with Action Features and Performance Evolution. In *Proceedings of the 5th International Conference on Educational Data Mining*, pages 57–64, 2012.
- [22] J. S. Kinnebrew, K. M. Loretz, and G. Biswas. A Contextualized, Differential Sequence Mining Method to Derive Students’ Learning Behavior Patterns. *Journal of Educational Data Mining*, 5(1):190–219, May 2013.
- [23] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 170–179, 2013.
- [24] S. Klingler, T. Käser, and B. Solenthaler. Temporally Coherent Clustering of Student Data. In *Proceedings of the 9th International Conference on Educational Data Mining*, pages 102–109, 2016.
- [25] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart. Predicting MOOC Dropout over Weeks Using Machine Learning Methods. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

- [26] V. Kovanović, D. Gašević, S. Dawson, S. Joksimović, R. S. Baker, and M. Hatala. Penetrating the black box of time-on-task estimation. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 184–193, 2015.
- [27] M. Köck and A. Paramythis. Activity sequence modelling and dynamic clustering for personalized e-learning. *User Modeling and User-Adapted Interaction*, 21(1-2):51–97, Apr. 2011.
- [28] X. Li, T. Wang, and H. Wang. Exploring N-gram Features in Clickstream Data for MOOC Learning Achievement Prediction. In *Database Systems for Advanced Applications*, 2017.
- [29] X. Li, L. Xie, and H. Wang. Grade prediction in MOOCs. In *2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, pages 386–392, 2016.
- [30] M. Liz-Dominguez, M. Caeiro-Rodriguez, M. Llamas-Nistal, and F. Mikic-Fonte. Predictors and Early Warning Systems in Higher Education - A Systematic Literature Review. In *LASI-SPAIN*, 2019.
- [31] L. P. Macfadyen and S. Dawson. Mining LMS data to develop an “early warning system” for educators: A proof of concept. *Computers & Education*, 54(2):588–599, Feb. 2010.
- [32] C. G. Marquardt, K. Becker, and D. D. A. Ruiz. A pre-processing tool for Web usage mining in the distance education domain. *Proceedings. International Database Engineering and Applications Symposium*, pages 78–87, 2004.
- [33] R. Martinez, K. Yacef, and J. Kay. Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In *Proceedings of the 4th International Conference on Educational Data Mining*, pages 111–120, 2011.
- [34] D. Mimno, H. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing Semantic Coherence in Topic Models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 262–272, 2011.
- [35] P. Mukala, J. J. Buijs, and V. d. Aalst. Exploring students’ learning behaviour in MOOCs using process mining techniques. In *BPM reports; Vol. 1510*, 2015.
- [36] M. Munk and M. Drlík. Impact of Different Pre-Processing Tasks on Effective Identification of Users’ Behavioral Patterns in Web-based Educational System. *Procedia Computer Science*, 4:1640–1649, Jan. 2011.
- [37] Z. A. Pardos and L. Horodyskyj. Analysis of Student Behaviour in Habitable Worlds Using Continuous Representation Visualization. *Journal of Learning Analytics*, 6(1):1–15, 2019.
- [38] J. Park, K. Denaro, F. Rodriguez, P. Smyth, and M. Warschauer. Detecting changes in student behavior from clickstream data. In *Proceedings of the Seventh International Learning Analytics and Knowledge Conference*, pages 21–30, 2017.
- [39] B. K. Pursel, L. Zhang, K. W. Jablow, G. W. Choi, and D. Velegol. Understanding MOOC students: motivations and behaviours indicative of MOOC completion. *Journal of Computer Assisted Learning*, 32(3):202–217, 2016.
- [40] A. Sheshadri, N. Gitinabard, C. F. Lynch, T. Barnes, and S. Heckman. Predicting Student Performance Based on Online Study Habits: A Study of Blended Courses. In *Proceedings of the 11th International Conference on Educational Data Mining*, pages 401–410, 2018.
- [41] B. Shih, K. Koedinger, and R. Scheines. Unsupervised Discovery of Student Strategies. In *Proceedings of the 3rd International Conference on Educational Data Mining*, pages 201–210, 2010.
- [42] T. Sinha, P. Jermann, and P. Dillenbourg. Your click decides your fate: Inferring Information Processing and Attrition Behavior from MOOC Video Clickstream Interactions. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 3–14, 2014.
- [43] K. Stevens, P. Kegelmeyer, D. Andrzejewski, and D. Buttler. Exploring Topic Coherence over Many Models and Many Topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 952–961, July 2012.
- [44] M. Wen and C. P. Rose. Identifying Latent Study Habits by Mining Learner Behavior Patterns in Massive Open Online Courses. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1983–1986, 2014.
- [45] A. F. Wise, J. Speer, F. Marbouti, and Y.-T. Hsiao. Broadening the Notion of Participation in Online Discussions: Examining Patterns in Learners’ Online Listening Behaviors. *Instructional Science: An International Journal of the Learning Sciences*, 41(2):323–343, Mar. 2013.

Unsupervised Approach for Modeling Content Structures of MOOCs

Fareedah ALSaad^{*}
University of Illinois at Urbana-Champaign
alsaad2@illinois.edu

Abdussalam Alawini
University of Illinois at Urbana-Champaign
alawini@illinois.edu

ABSTRACT

With the increased number of MOOC offerings, it is unclear how these courses are related. Previous work has focused on capturing the prerequisite relationships between courses, lectures, and concepts. However, it is also essential to model the content structure of MOOC courses. Constructing a precedence graph that models the similarities and variations of learning paths followed by similar MOOCs would help both students and instructors. Students can personalize their learning by choosing the desired learning path and lectures across several courses guided by the precedence graph. Similarly, by examining the precedence graph, instructors can 1) identify knowledge gaps in their MOOC offerings, and 2) find alternative course plans. In this paper, we propose an unsupervised approach to build the precedence graph of similar MOOCs, where nodes are clusters of lectures with similar content, and edges depict alternative precedence relationships. Our approach to cluster similar lectures based on PCK-Means clustering algorithm that incorporates pairwise constraints: Must-Link and Cannot-Link with the standard K-Means algorithm. To build the precedence graph, we link the clusters according to the precedence relations mined from current MOOCs. Experiments over real-world MOOC data show that PCK-Means with our proposed pairwise constraints outperform the K-Means algorithm in both Adjusted Mutual Information (AMI) and Fowlkes-Mallows scores (FMI).

Keywords

Precedence Graph, Clustering, Pairwise Constraints, Precedence Relations, Alternative Learning Paths, Common Learning Path.

1. INTRODUCTION

According to Class Central [19], by the end of 2019, over 13 thousand MOOCs have been announced or launched by more than 900 universities worldwide. With such an increase in online courses, it becomes increasingly hard for learners to understand similarities and differences among courses that cover similar topics. For instance, Coursera¹, one of the leading MOOC platforms, offers

^{*}King AbdulAziz University, Jeddah, Saudi Arabia.

¹<https://www.coursera.org>

Fareedah Alsaad and Abdussalam Alawini "Unsupervised Approach for Modeling Content Structures of MOOCs" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 18 - 28

several "Machine Learning" courses, such as "Machine Learning" from Stanford University, "Machine Learning with Python" offered by IBM, "Machine Learning for All" from University of London, etc. Understanding the content structure across such similar courses can be very challenging. Consequently, MOOCs users may waste time choosing a course among a broad set of similar MOOC offerings.

Previous work studied ways for capturing prerequisite relationships between courses [23, 11], between lectures within (or among) courses [5, 6], or between concepts discussed within (or across) courses [2, 10, 15, 23, 11]. While modeling prerequisite relationships is crucial for understanding the content and knowledge structure of a specific domain, prerequisites do not reveal content overlap in similar courses. Further, modeling MOOC content in terms of prerequisite relations cannot detect the variations in the learning path between similar MOOCs.

In this paper, we propose to model the content structure of similar MOOC offerings as a precedence graph. This graph can be useful for both learners and instructors. Learners can use the graph to build a customized learning plan as well as to explore how various courses explain the same topic. As for instructors, the graph can be used to identify any missing knowledge in their MOOCs offering, hence help them improve their courses. Section 3.2 elaborates on other possible applications of our proposed MOOCs precedence graph.

More precisely, we introduce an unsupervised approach to model the content structures of MOOCs. Figure 1 demonstrates the proposed idea. Given a set of courses that have some overlap in their content, we first cluster lectures based on their content similarity into clusters; each cluster represents a node in the precedence graph (see Figure 1 (b)). Then, the clusters are linked according to their lectures precedence relations mined from current MOOCs as depicted in Figure 1 (c). Linking clusters of similar content based on the precedence relations can reveal the various possible paths followed by similar courses and also capture which path is considered more common in these courses.

To cluster lectures based on their content similarity, we utilize a constraint-based clustering algorithm called Pairwise Constrained K-Means (PCK-Means). PCK-Means guides the clustering process by using two constraints: Must-Link and Cannot-Link. The idea is to guide the clustering process, by using the constraints, to focus on clustering lectures across courses instead of within courses to capture the similarity between courses. To measure the content similarity between lectures, we exploit both lecture titles and transcripts as they both encode enough information about the content of lectures. By using cosine similarity, we measure the similarity between lectures

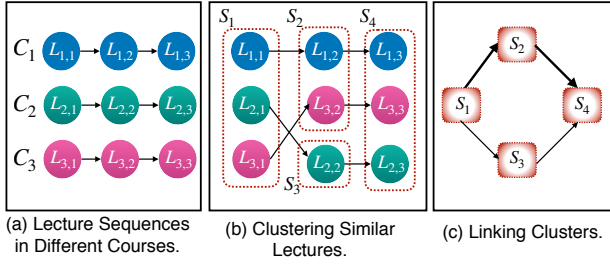


Figure 1: The basic idea of modeling the content of MOOCs to construct the precedence graph. Given similar Courses with some overlaps in content represented as sequences of lectures, the precedence graph is constructed by clustering lectures based on content similarity and link the clusters using the precedence relations between lectures.

and construct the constraint examples to guide the clustering process. Our experiment on real MOOC dataset shows that PCK-Means with our proposed constraints outperforms standard K-Means algorithm in both Adjusted Mutual Information (AMI) and Fowlkes-Mallows scores (FMI).

After clustering similar lectures, we construct the precedence graph by linking clusters based on the precedence relations and label clusters using salient and key terms in each cluster. The generated precedence graph reveals the popular learning path and some alternative paths in our MOOCs dataset.

The rest of the paper is organized as follows. Section 2 presents related work. In section 3, we demonstrate the idea of modeling the content structure of MOOCs by an illustrative example and also present some applications of the precedence graph before we formally define our problem in Section 4. Section 5 describes how we represent the content of MOOCs using word count and embedding representations. In section 6, we explain PCK-Means algorithm and present our method of generating the lists of pairwise constraints. In section 7, we demonstrate the process of linking and labeling clusters to construct the precedence graph. Section 8 elaborates on our approach for the evaluation and presents some learning path examples extracted from the generated precedence graph. Finally, we conclude our work in section 9.

2. RELATED WORK

There has been recently a growing body of work that addresses the problem of modeling the content of MOOCs. Most of this work has focused on capturing the prerequisite relationships between courses [23, 11], between lectures or segments of lectures [5, 6], or between concepts discussed within or across courses [2, 10, 15, 23, 11]. These studies have developed supervised and unsupervised approaches to model only the prerequisite relations in MOOCs. In this paper, we go further and develop an unsupervised approach to capture the similarities and variations of learning paths between MOOCs in the same domain. Our work models the precedence relations (i.e., the implicit prerequisite relationships) between concepts by clustering similar lectures among different courses. Therefore, our model can reveal popular learning paths shared by several courses along with alternative possible paths to learn the topic covered by these similar courses.

To model the prerequisite relationships, some studies have used external knowledge such as Wikipedia to support identifying educational concepts [10] or to represent concepts using Wikipedia articles or categories [15, 23, 11]. Using Wikipedia to identify concepts has some weaknesses: (1) some concepts are not included in Wikipedia [15] and thus can affect the performance of the model, (2) the mapping between course concepts to Wikipedia is not always accurate, which can affect the quality of the extracted concepts [10], and (3) using Wikipedia categories affects concept granularity by preferring more general concepts [2]. Instead of using Wikipedia, the work by ALSaad et al. [2] has exploited pre-trained part-of-speech-guided phrasal segmentation to extract phrases from course content and then manually group synonym phrases to represent concepts. In our work, instead of relying on external knowledge or manually improve the concepts, we represent the precedence graph nodes by salient terms using simple TF-IDF and bag-of-words representations. Our method represent each cluster with key terms by accumulating lecture representation vectors of each cluster and exploiting the top ranked words to represent clusters. Accumulating the vector representations of similar lectures helps in extracting representative terms that express the content of each cluster clearly.

Another related line of work is the use of prerequisite relations between concepts to organize learning units and predict the precedence relationships between them [1, 13]. The studies [1, 13] have proposed supervised approaches that rely on features extracted from external knowledge such as Wikipedia [1] and DBpedia [13] to infer the prerequisite relations between concepts. While the work [1] assumed that concepts are given, the study [13] manually extracted concepts by annotators. Our work is different as instead of inferring the prerequisite relations between concepts and then organizing them according to the precedence relations, we leverage the precedence relations between lectures in existing MOOCs to detect the precedence relations between the nodes in the precedence graph. Each node in the precedence graph is labeled automatically with key concepts that clearly express the content of each node without the use of external knowledge.

The work by Shah et al. [20] is the most relevant work to ours. The study has proposed a method for linking similar courses to construct a map of lectures connected by two types of relations: similar and prerequisite. The goal of the map is to help students find the desired learning path that fits their interests and backgrounds. Our work is very similar as we construct the precedence graph that depicts the different possible learning paths. However, instead of linking lectures by similar and prerequisite relations, we cluster lectures based on content similarity and connect clusters according to the precedence relations. Our approach reveals the similarities and variations of learning paths between different courses by capturing popular learning paths shared by many courses in the domain, hence emphasizes the importance of the common, comprehensive and alternative learning paths.

3. MODELING MOOCs CONTENT

In this section, we explain the idea of modeling the content of MOOCs as a precedence graph by using an illustrative example. We also discuss possible applications of the mined precedence graph.

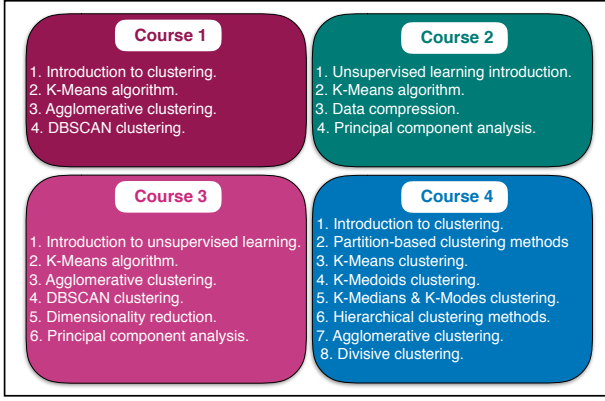


Figure 2: The sequence of lectures in four different courses that explain the topic Unsupervised Learning.

3.1 An Illustrative Example

For illustration purposes, let us assume that a MOOC platform offers four courses about *Unsupervised Learning* topic in machine learning as shown in Figure 2. Each course explains the topic using a sequence of lectures. As can be seen in Figure 2, there are some overlaps between these four courses as they all teach the same topic, but there are also some variations. The variation in each course is based on instructors' perspectives and background about the topic, instructors' teaching styles, and also the learning objective of each course. Some courses are abstract as they focus on the theory behind the topic while other courses are more concrete as they demonstrate the topic by illustrating real-world examples. Courses also vary in the coverage of topics as some courses are concise while other courses cover topics in more details. For example, **Course 1** and **Course 2** in Figure 2 are examples of concise courses that focus only on teaching the main concepts in the topic. In contrast, **Course 3** and **Course 4** are examples of courses that elaborate more in the topic by providing more detailed concepts.

Given the similarities and variations between these courses that explain the same subject, we investigate the following questions. *how these courses are related? What are the common concepts taught by the majority of these courses? Is there a common learning path shared by most of these courses? what are the alternative paths to study the topic?* Modeling the content structure of these courses as a precedence graph is a crucial step to help learners and educators with answering these questions.

The first step in building the precedence graph is to cluster lectures based on their content similarity and then construct a node in the graph for each cluster. Figure 3 shows the cluster assignment of each course lecture of Figure 2. As illustrated in Figure 3, all the introductory lectures, the first lecture of each course, are grouped into one cluster (cluster S_1) as all these lectures introduce the topic of *Unsupervised Learning*. Similarly, all the lectures about the concepts “*K-Means Algorithm*”, “*Agglomerative Clustering*”, and “*DBSCAN*” are clustered into three different clusters: S_6 , S_3 , and S_7 respectively. Furthermore, lectures about “*Data Compression*” are clustered into cluster S_{11} while lectures taught “*Principal Component Analysis*” concept are clustered into cluster S_{10} .

After clustering similar lectures and finding the nodes of the precedence graph, the next step in building the graph is to link these

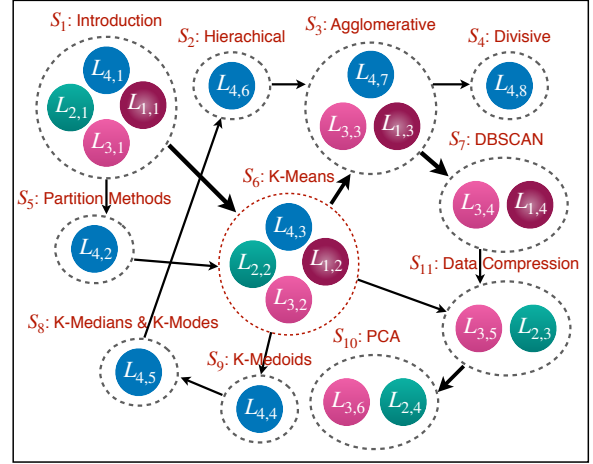


Figure 3: The mined precedence Graph from courses in Figure 2. Lectures are grouped into clusters to construct the nodes of the graph. Edges depict the precedence relationships between clusters where thick edges represent the edges with high weights and thus indicate how common are the relations between the nodes.

scattered clusters to reveal the precedence relations between clusters. To that end, we use the precedence relations between adjacent lectures of the same course to construct the edges between nodes (clusters) in the precedence graph. For instance, we add a directed edge from cluster S_5 to cluster S_6 to determine the precedence relation between these two nodes according to the sequence of lectures 2 and 3 in course 4. To reflect the strength of each precedence relations between two nodes (i.e., how common are the relations between the nodes), we attach each edge in the precedence graph with different weights. Edge weights are calculated by accumulating the frequency of lecture sequences in various courses. For example, as shown in Figure 3, the strength weight of edge $S_1 \rightarrow S_6$ should be higher than the strength weight of edge $S_1 \rightarrow S_5$ as three out of the four courses (1,2, and 3) have the sequence $S_1 \rightarrow S_6$ while only one course (4) shows the sequence $S_1 \rightarrow S_5$.

As mentioned earlier, the mined precedence graph can help us in revealing some hidden structures in similar MOOCs. For instance, it is clear from Figure 3 that the path $\{S_1 \rightarrow S_6 \rightarrow S_3 \rightarrow S_7\}$ is more common than other paths. The reason is that three courses (1,2, and 3) explain the concepts “*K-Means Algorithm*” after introducing the topic and two of them (courses 1 and 3) present the concepts “*Agglomerative Clustering*” and “*DBSCAN*” after that. In addition to indicating the common path, the mined precedence graph can also reveal other possible paths to learn the topic such as the path $\{S_1 \rightarrow S_5 \rightarrow S_6 \rightarrow S_3 \rightarrow S_4\}$, or the path $\{S_1 \rightarrow S_5 \rightarrow S_6 \rightarrow S_{11} \rightarrow S_{10}\}$. All these paths are valid and, off course, choosing a path depends on students' learning objectives.

In general, the mined precedence graph helps in capturing the similarities and variations of the learning paths of similar courses in our illustrative examples. In section 8.3, we present some learning path examples from the precedence graph generated by our approach.

3.2 Precedence Graph Applications

Our mined precedence graph can be used to support several applications for improving the learning and teaching process. However,

before discussing these applications, we first want to clarify that (in this paper), we define a student as a person who uses MOOCs as modularized resources to learn topics of their choice (as opposed to taking a full course as part of a certificate program.) According to Zheng et al. [24], one of the motivations for a student to register for a MOOC is to learn some desired concepts on-demand. Once they achieve their learning goals, this type of student usually stops participating in the course.

Our precedence graph can support the following applications.

Personalized (customized) course plans. Our precedence graph can help students develop custom learning plans. Students can examine the graph to identify possible alternative paths for learning a topic and then choose the path that best fits their needs. For instance, a student might choose to follow one of the following two paths: $\{S_6 \rightarrow S_9 \rightarrow S_8\}$ or $\{S_6 \rightarrow S_{11} \rightarrow S_{10}\}$ shown in Figure 3. The former path helps the student explore and learn about various clustering algorithms: “K-Means Algorithm”, “K-Medoids Algorithm”, and “K-Medians and K-Modes Algorithms”, while the latter path helps the student learn about the concepts of “Data Compression”, and “PCA” with “K-Means” clustering algorithm.

An overview/summary of a topic. There are two ways in which the precedence graph can be used to help students obtain a quick overview of a particular topic of interest. First, students can use the graph to follow the most common path that is shared among several courses (i.e., the path with the highest edge weights.) For instance, students can follow the path: $\{S_1 \rightarrow S_6 \rightarrow S_3 \rightarrow S_7\}$ as this is the path with the highest edge weights in the graph shown in Figure 3. This path introduces the topic of *Unsupervised Learning* first before presenting three important and well-known clustering algorithms: “K-Means Algorithm”, “Agglomerative Clustering Algorithm”, and “DBSCAN Algorithm”. Second, using summarization algorithms, we can generate a summary of the lectures in each node (cluster) of the most common path in the precedence graph. Such a succinct representation of clusters would provide students with a concise summary of the topic they want to learn.

Acquiring expert knowledge. Our precedence graph can also be used by students who are interested in becoming experts in a particular domain. The graph allows students to easily determine how the knowledge of a domain is structured. It also allows them to choose the path that exposes them to a variety of concepts related to the topic they want to learn. For example, to learn the most about unsupervised learning, a student can follow the longest path in the precedence graph shown in Figure 3: $\{S_1 \rightarrow S_5 \rightarrow S_6 \rightarrow S_9 \rightarrow S_8 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_7 \rightarrow S_{11} \rightarrow S_{10}\}$. Clustering similar lectures from various courses into the same clusters can also help this type of students as they can explore how different courses explain the same concept.

Helping instructors improve their courses. In addition to helping students with their learning process, the mined precedence graph can also aid instructors in understanding the structure of their MOOC offerings. By examining the precedence graph, instructors can identify potential knowledge gaps (missing topics) or a better ordering of the topics, and hence incorporate the new knowledge in their next course offerings.

4. PROBLEM DEFINITION

The design of a MOOC mimics that of a typical on-campus course in which the fundamental structure is a sequence of lectures. By

leveraging the sequences of lectures and the content similarities between lectures from similar courses, we can model the knowledge structure of similar (i.e., courses that cover the same topic) MOOCs as a precedence graph. The nodes of this graph are groups of similar lectures, labeled by dominant and salient terms in these lectures. The edges of the graph represent the alternative precedence relations between nodes. Each edge can be assigned different weights that reflect the strength of the relation.

We formally define the problem as follows. Given a set of courses $X = \{C_1, C_2, C_3, \dots, C_n\}$, where n is the total number of courses. We assume that all courses in X have the same difficulty level, and there are some content overlaps between courses. Each course C_i is an ordered list of lectures $C_i = [L_{i1}, L_{i2}, \dots, L_{i|C_i|}]$, where $|C_i|$ is the total number of lectures in the course C_i . Each lecture L_{ij} is represented using the title t_{ij} and the lecture transcript d_{ij} . The goal is to model the content structure of similar MOOCs by constructing the precedence graph as a directed graph $G = (V, E)$ where V is the set of nodes, $V = \{S_1, S_2, S_3, \dots, S_{|V|}\}$ (the number of nodes $|V|$ is given), and $E = \{e_1, e_2, e_3, \dots, e_{|E|}\}$ is the set of edges between nodes. Edges in the graph G are directed edges to indicate the precedence relations between nodes. Each node in the precedence graph is a cluster or a group of lectures that have similar content. For example, $S_v = \{L_{i1}, L_{i2}, L_{j5}\}$ is a cluster that has the first two lectures from course C_i and the fifth lecture from course C_j . We represent the precedence graph G as an edge weight matrix $\mathbf{G} \in \mathbb{R}^{|V| \times |V|}$ where each entry of matrix \mathbf{G} contains the edge weight. For instance, the edge weight of the entry g_{ij} reflects the strength of the precedence relationship from cluster S_i to cluster S_j .

To construct the precedence graph, we need first to find the set of nodes V of the precedence graph by grouping similar lectures using both lecture titles t_{ij} and lecture transcripts d_{ij} . Then, we compute the edge weight between pairs of nodes by leveraging the sequence of lecture in each course C_i . Sections 6 and 7 explain our proposed approach to build the precedence graph.

5. MOOC CONTENT REPRESENTATION

In this section, we demonstrate how we represent MOOC lectures by exploiting two representations: 1) *sparse representation* that is based on word count, and 2) *dense representation* to capture the semantic similarity between text. The purpose of using these two representations is to compare how each of them affects the performance of clustering.

To represent lectures, we use the sparse representation, a robust and straightforward representation based on the count of words. We represent lecture titles as vectors of word count using Bag-Of-Words (BOW) representation. Since lecture titles are short and concise, the frequency of each word in the BOW vector is usually one. The bag-of-words representation can be thought of as a bit vector where a bit is set to 1 when the word occurs in the title and set to 0, otherwise.

For representing lecture transcripts, we use the Term-Frequency Inverse-Document-Frequency (TF-IDF) representation. TF-IDF weighting takes into consideration the count of words in documents as well as the popularity of words in the corpus, hence gives higher weights to the words that are more frequent in the document and less popular in the corpus.

In our model, each lecture is represented by two vectors: a BOW vector to represent the title and a TF-IDF vector to capture the content of the transcript. The drawbacks of this representation are 1)

it generates high dimensional sparse vectors, and 2) it cannot capture the semantic similarity between similar words.

To overcome the limitations of TF-IDF and BOW representations, we use an alternative (dense) representation to model MOOC content: the unsupervised smoothed inverse frequency (uSIF) [8]. The uSIF is a simple, yet effective method for generating sentence embeddings without any labeled data. It is an improvement of smoothed inverse frequency (SIF) [18], one of the state-of-the-art embedding representation for longer pieces of text. The basic idea of uSIF is to exploit the pretrained word embeddings such as Word2Vec [14] or Glove [16] that capture the semantic meaning between words to learn the embeddings of sentences and paragraphs taking into consideration the frequency of words in the text. For more information about uSIF, please refer to [8].

For the embedding representation of lectures, we use uSIF with Glove pretrained word embeddings [4] to represent both lecture titles and transcripts. The number of feature dimensions in embedding vectors is 100 dimensions.

6. CLUSTERING LECTURES OF MOOCS

To construct the nodes of the precedence graph, lectures are grouped into clusters based on their content similarity. We can use any clustering algorithm such as K-Means to do the clustering of lectures. However, one problem of using K-Means or some other clustering algorithms is that they will cluster similar lectures not just across courses but also within courses. For example, if one course explains the topic “Gradient Descent in Logistic Regression” and then later explains the topic “Gradient Descent in Neural Networks”, then there is a high chance that the clustering algorithm would group these two lectures into the same cluster as the course instructor would use almost the same terminology to explain these two topics. However, our goal is to capture the similarity of lectures across courses to reveal common learning paths utilized by many courses as well as other alternative learning paths. Therefore, we need to restrict the clustering process to cluster lectures from different courses rather than within the same course. To do that, we need to guide the clustering algorithm by imposing some constraints; which is infeasible with the standard K-Means algorithm. Therefore, we decided to exploit a constraint-based clustering algorithm called Pairwise Constrained K-Means (PCK-Means) [3] to guide the clustering process.

6.1 PCK-Means Clustering Algorithm

PCK-Means clustering algorithm [3] is a variation of the standard K-Means algorithm that incorporates distance between points as well as pairwise constraints to guide the clustering process. PCK-Means is a semi-supervised approach where users provide some labels or pairwise constraints that the algorithm uses to improve the clustering. Since collecting labels from users is expensive, we propose an unsupervised method by automatically find suitable labels or constraints to guide the clustering process (discussed in section 6.2).

Pairwise constraints can be used to determine the prior knowledge about the domain by specifying which instances (in our case lectures) should or should not be clustered together [21, 3]. There are two types of pairwise constraints: **Must-Link** and **Cannot-Link**. Must-Link constraint specifies pairs of instances (lectures) that need to be grouped into the same cluster, while Cannot-Link constraint determines pairs that should not be in the same cluster. Each type of pairwise constraint applies a penalty function when the constraint is violated. The objective function of PCK-Means is to 1) choose

partitions that minimize the penalty cost of each constraint, and 2) minimize the sum of the square distance between the points and the centroids of the clusters they belong to.

More formally, let \mathcal{M} be a list of Must-Link constraint, which includes tuples of lectures (L_i, L_j) that needs to be clustered together. Let \mathcal{C} be a list of Cannot-Link constraint. Each item in \mathcal{C} is a lecture pair of the form (L_i, L_j) where lecture L_i and L_j should not be in same cluster. Each tuple in \mathcal{M} and \mathcal{C} is order-independent. Assume $W = \{w_{i,j}\}$ and $\bar{W} = \{\bar{w}_{i,j}\}$ are the sets of penalty costs of violating the **Must-Link** and **Cannot-Link** constraints respectively. Each lecture L_i is assigned to a cluster \mathcal{S}_i , where $\mathcal{S}_i \in \{h\}_{h=1}^{|V|}$, by minimizing both the distance between L_i and the cluster centroid $\mu_{\mathcal{S}_i}$ and the penalty costs of violating the constraints. The objective function of PCK-Means algorithm is as follow:

$$\begin{aligned} \mathcal{J}_{pckm} = & \frac{1}{2} \sum_{L_i \in X} \|L_i - \mu_{\mathcal{S}_i}\|^2 \\ & + \sum_{(L_i, L_j) \in \mathcal{M}} w_{ij} \mathbb{1}[\mathcal{S}_i \neq \mathcal{S}_j] + \sum_{(L_i, L_j) \in \mathcal{C}} \bar{w}_{ij} \mathbb{1}[\mathcal{S}_i = \mathcal{S}_j] \quad (1) \end{aligned}$$

The first part of the objective function is K-Means objective function while the second and the third parts are the accumulated penalty costs of violating the Must-Link and Cannot-Link constraints respectively. The $\mathbb{1}[\cdot]$ is the indicator function where $\mathbb{1}[true] = 1$ and $\mathbb{1}[false] = 0$.

In the initialization step of PCK-Means, examples of the pairwise constraints are used to estimate the centroids of clusters. Before initializing the cluster centroids, PCK-Means finds the transitive closure of tuples in Must-Link constraint and appends them to the list of Must-Link constraints. Then the updated list is used to create λ neighborhood sets. For each pair of neighborhoods, P_i and P_j with at least one pair of points that appear in the Cannot-Link list, PCK-means generates Cannot-Link constraint tuples between every pair of points in P_i and P_j and appends these tuples to the Cannot-Link constraints. Then the algorithm gets λ neighborhoods where links of type Must-Link constraint connect points within each neighborhood, and links of type Cannot-Link constraint connect some neighborhoods. If λ is higher than the number of clusters, $\lambda > |V|$, then the algorithm chooses the neighborhood sets with the largest number of instances to initialize the clusters and the centroids of each cluster. In contrast, if λ is less than the number of clusters, $\lambda < |V|$, then PCK-Means initializes the clusters from the λ neighborhoods and looks for a point that has links of type Cannot-Link constraint to all the λ neighborhoods. If so, it initializes a new $\lambda + 1$ cluster from this point. Otherwise, PCK-Means chooses the remaining $|V| - \lambda$ clusters randomly.

In general, the PCK-Means clustering algorithm is an iterative algorithm where it starts by using the pairwise constraints to initialize the clusters. Then, iteratively (1) assign points (or lectures) to clusters that minimize the combined objective function and then (2) re-estimate the centroids of each cluster according to the cluster assignment of each point. These two steps are repeated until the algorithm converges. For more information about the algorithm, please refer to [3].

6.2 Pairwise Constraints

To build the precedence graph, we use Must-Link and Cannot-Link constraints to guide the clustering process. Must-Link constraint

includes a list of pairs of lectures that have higher chance to be similar while Cannot-Link constraint contains a list of lecture pairs that have lower chance to be part of the same clusters such as lectures from the same course. Yet, the question is how to find good examples of lecture pairs for the lists of Must-Link and Cannot-Link constraints.

6.2.1 Must-Link Constraint

As our goal is to capture the content similarity between lectures across courses, we want to feed the algorithm with similar lectures from different courses that have higher chance to be part of the same cluster as examples of Must-Link instances. To do that, we can use the cosine similarity measure to calculate the similarity score between lectures from different courses and choose lecture pairs with a similarity score exceeds some predefined threshold.

Besides Similar lectures across courses, some similar lectures within the same course can be good examples of Must-Link instances. Adjacent lectures can have very similar content and hence they should be grouped together in the same cluster. For instance, the two adjacent lectures “K-Means Algorithm” and “Initialization of K-Means Clustering” have similar content as they talked about *K-Means Clustering Algorithm* and thus they need to be grouped together. Therefore, we add adjacent lectures that have a similarity score greater than the predefined threshold.

We propose two approaches to capture the similarity between lectures within courses or among courses. First, we use the cosine similarity between two lectures represented by lecture transcripts. Pairs of lectures are considered similar when they have similar content and hence the cosine similarity score would be high. Second, we use the cosine similarity between two lectures represented by lecture titles. We believe that two lectures are similar when they have very similar titles even when there are some variations in the content. One reason is that instructors sometimes explain the topic from different perspectives. For instance, one instructor might explain the lecture with a title “K-Means Clustering Algorithm” by using examples while another instructor might explain the same lecture by illustrating the theory behind it. Although the content is different, both lectures explain the same topic but from different perspectives. Another reason of using lecture titles is due to the average length of lectures in MOOCs. Lectures in MOOCs are usually shorter in length compared with regular university classes. As a result, some instructors split the topic into two or more lectures. Usually these lectures have very similar titles and should be clustered together even if their content might vary. Therefore, we decided to utilize lecture titles to measure the similarity between lectures in addition to lecture transcripts. However, we use two different predefined thresholds K_1 and K_2 to capture the lectures similarity using titles and transcripts respectively as we have to set a higher threshold for titles to minimize the noise.

In general, the list of Must-Link constraint contains any similar lectures across courses and similar adjacent lectures within courses.

6.2.2 Cannot-Link Constraint

Unlike the Must-Link constraint, Cannot-Link constraint is used to indicate lecture pairs that should not be part of the same clusters. Since we want to force the clustering algorithm to capture the similarity between lectures across courses, we add lecture pairs from the same course to the list of Cannot-Link constraint. However, not any pair can be added to the list as some adjacent lectures can have similar content or similar titles and hence need to be grouped into the same cluster. Therefore, to determine lecture pairs that are

suitable to be examples of Cannot-Link constraint, we apply the cosine similarity on the transcripts of two adjacent lectures. When the cosine similarity of two adjacent lectures, L_{ij} and $L_{i(j+1)}$ of course C_i , are less than a predefined threshold K_3 , then we can say that there is a **topic shift** and hence we can add these two adjacent lectures to the list of Cannot-Link constraint. However, before adding any lecture pairs to the list of Cannot-Link constraint, we need to ensure that the pair is not part of the Must-Link constraint and its transitive closure list. In addition to adding the two adjacent lectures L_{ij} and $L_{i(j+1)}$, we also pair the lecture L_{ij} with all the subsequent lectures of lecture $L_{i(j+1)}$ since there is a shift in the topic. As a result, we add the lectures $(L_{ij}, L_{i(j+z)})$ where $1 < z < |C_i| - j$, to the list of Cannot-Link constraint if they are not part of the Must-Link constraint and its transitive closure list.

In general, the purpose of Cannot-Link constraint is to restrict the clustering algorithm from clustering lectures within courses in order to capture the similarity between different courses. As a result, by using Must-Link and Cannot-Link constraints, the clustering algorithm learns to cluster lectures from across courses and only cluster adjacent lectures within the same course if they are similar.

7. BUILDING PRECEDENCE GRAPH

Building the precedence graph from similar MOOCs has three steps: (1) Cluster similar lectures to construct the node of the graph, (2) Link the nodes by a directed weighted edge to determine the precedence relations between nodes, and (3) Represent each node by dominant and salient terms mined from lectures belong to each nodes. In the previous section, we explain how we cluster similar lectures using PCK-Means algorithm with our proposed Must-Link and Cannot-Link constraints. In this section, we first present our method of linking the precedence graph nodes before illustrating our approach of labeling each node.

7.1 Linking Clusters

After clustering similar lectures, we need to link the scattered clusters to construct the precedence graph. As we mentioned earlier, we utilize lecture sequences in each course. We can think of the sequence of lectures in MOOCs as implicit prerequisite relationships between lectures as these sequences are carefully designed by experts. When instructors design courses, they usually maintain the prerequisite order constraints between lectures by placing prerequisite lectures before the dependent lectures. In addition, according to the *locality of references* property [1], when designing a course plan, a dependent lecture should appear as soon as possible after the prerequisite lecture to reduce students comprehension burden. Therefore, tackling the various sequence orders of lectures from different courses helps in linking clusters of lectures from across courses and thus captures the precedence relations between clusters.

To link the scattered clusters, we use the precedence relations between adjacent lectures to infer the precedence relations between clusters. If two adjacent lectures L_{ij} and $L_{i(j+1)}$ of course C_i appear in two different clusters, then these two clusters need to be linked by an edge with a direction from the cluster that includes lecture L_{ij} to the cluster that has lecture $L_{i(j+1)}$. Sometimes some adjacent lectures appear in the same cluster and hence we ignore the sequence relation of these lectures.

To capture the strength of the precedence relations between clusters, and hence how these relations are common in current MOOCs, we attach each edge with different weights. We accumulate the frequency of courses that have adjacent lectures clustered into two different

clusters to determine the weight between these two clusters. The equation to determine the edge weight is as follow:

$$W(\mathcal{S}_i \rightarrow \mathcal{S}_j) = \sum_{\forall C \in X} \sum_{z=1}^{|C|-1} \mathbb{1}[L_z \in \mathcal{S}_i \wedge L_{z+1} \in \mathcal{S}_j] \quad (2)$$

where $W(\mathcal{S}_i \rightarrow \mathcal{S}_j)$ is the weight of the edge between cluster \mathcal{S}_i and \mathcal{S}_j and $\mathbb{1}[\cdot]$ is an indicator function where $\mathbb{1}[true] = 1$ and $\mathbb{1}[false] = 0$.

Since the edge weights determine the popularity of relations across similar courses, edge weights are not normalized to be between 0 and 1 because normalization will produce misleading weights. For example, if we use normalized edge weights, then the edge that connects two clusters that have adjacent lectures from one course will have the same weight; which is equal to 1, to the edge that connects two clusters that have adjacent lectures from N courses. Therefore, we use unnormalized edge weights to capture the popularity of the precedence relations.

7.2 Labeling Clusters

Each node in the precedence graph is labeled by some key terms to represent the topics or key concepts discussed by the lectures attached to this node. To extract the key terms from lectures, we exploit lecture titles and transcripts represented by bag-of-words and TF-IDF representations respectively. Lecture titles are very concise and usually have the key terms in lectures. On the other hand, lecture transcripts are more elaborative and would help in extracting other important key terms that demonstrate topics or key concepts of each cluster.

The basic idea to extract the key terms is to accumulate the vector representations of each lecture that belongs to the same cluster in order to find the key terms of that cluster. In other words, for all lectures that belong to the same cluster we accumulate the bag-of-word representation vectors of their titles and also add the TF-IDF weighting vectors of their transcripts. Then, we use the top k terms from these two different representations to find the salient terms that represent each cluster. The following is the equation used to specify the key words of each cluster:

$$Label(\mathcal{S}_i) = \left(\max_k \sum_{j=1}^{|D|} \sum_{\forall L \in \mathcal{S}_i} TFIDF(w_j | w_j \in d_L) \right) \cup \left(\max_k \sum_{j=1}^{|T|} \sum_{\forall L \in \mathcal{S}_i} BOW(w_j | w_j \in t_L) \right) \quad (3)$$

where the first part finds the top k terms by using the TF-IDF representation of lecture transcripts d_L where $|D|$ is the total number of vocabularies in the corpus of lecture transcripts. For each word w_j in the vocabulary, we accumulate the TF-IDF weights of word w_j if the word appears in lecture L that belongs to cluster \mathcal{S}_i . Similarly, the second part determines the top k terms by exploiting the bag-of-words representation of titles t_L where the total number of vocabularies in lecture titles is $|T|$. We also accumulate the BOW weights of each word belongs to titles of all lectures that are part of

cluster \mathcal{S}_i . By taking the union of these two sets of top words, we extract salient terms that clearly explain the content of each clusters.

8. EVALUATION

In this section, we evaluate the performance of our approach for clustering similar lectures using PCK-Means algorithm with the proposed pairwise constraints. We first present the dataset and ground truth we used in our evaluation. Then, we compare the performance of the clustering algorithms using both representations: word counts (sparse representation) and embeddings (dense representation). We also present some examples of the learning paths extracted from the precedence graph that was constructed by our approach. Finally, we discuss some limitations of our study.

8.1 Datasets

We used a dataset of six modules related to *Unsupervised Learning* and *Clustering Algorithms* from five real machine learning and data mining courses offered by the Coursera platform². These modules include “Unsupervised Machine Learning”, “Partitioning Based Clustering Methods and Hierarchical Clustering Methods”, “Unsupervised Learning”, “Clustering”, “Clustering With K-Means”, and “Hierarchical Clustering” (see Table 1.) The total number of lectures in the dataset is 65 lectures. Each lecture is represented by its title and transcript.

To evaluate the performance of the PCK-Means algorithm and the effectiveness of the proposed constraints, we asked experts to construct the ground truth labels of our dataset. Each of our four experts (a Machine Learning professor, an Information Science professor, a Machine Learning graduate student, and a Database and Information Systems graduate student) manually grouped lectures based on topics similarities. None of the experts is participating in this study.

To measure the level of agreement among our experts, we used the Fleiss’ kappa measure. Fleiss’ Kappa is a statistical measure of inter-rater agreement used to determine the level of agreement between two or more raters. The kappa score of labels collected from experts was $\kappa = 0.65$, which indicates substantial agreements between the annotators.

After receiving the labeled datasets from our experts, we used the majority votes to decide the cluster assignment of each lecture. For lectures that experts disagreed on their clustering assignment, we decided to follow the advice of our experts and created a new cluster for each lecture. The total number of labeled clusters was 21 clusters.

8.2 Clustering Performance

To evaluate the performance of our clustering approach and to study the effect of using the pairwise constraints on clustering performance, we compared the PCK-Means algorithm to the standard K-Means algorithm. In particular, we focused on two measures: (1) Adjusted Mutual Information (AMI), and (2) Fowlkes-Mallows scores (FMI). Adjusted Mutual Information is a variation of the Mutual Information measure that is used for comparing clustering results. According to

²<https://www.coursera.org>

³<https://www.coursera.org/learn/advanced-machine-learning-signal-processing>

⁴<https://www.coursera.org/learn/cluster-analysis>

⁵<https://www.coursera.org/learn/machine-learning>

⁶<https://www.coursera.org/learn/machine-learning-with-python>

⁷<https://www.coursera.org/learn/ml-clustering-and-retrieval>

Table 1: The dataset utilized for the evaluation. It has six modules from five courses. The total number of lectures in the dataset is 65 lectures.

Courses	Modules	# of Lectures
Advanced Machine Learning and Signal Processing ³	Unsupervised Machine Learning	13
Cluster Analysis in Data Mining ⁴	Partitioning Based Clustering Methods and Hierarchical Clustering Methods	15
Machine Learning ⁵	Unsupervised Learning	12
Machine Learning With Python ⁶	Clustering	6
Machine Learning Clustering and Retrieval ⁷	Clustering With K-Means	13
	Hierarchical Clustering	6

Romano et al. [17], AMI measure should be used to evaluate the clustering performance when the reference clustering is unbalanced and contain small clusters. Since we have unbalanced clusters (i.e., some clusters have many lectures while others have one or two lectures), we decided to use AMI for the evaluation. The second metric, Fowlkes-Mallows scores, is a geometric mean of precision and recall where precision determines the correctness of the clustering assignments of lectures while recall measures the completeness of the assignments. Similar to AMI, FMI gives a zero score for random clustering assignments.

Before discussing clustering performance, it worth mentioning that for finding the lists of Must-Link and Cannot-Link constraints, we tried various values for each threshold, K_1 , K_2 , and K_3 , and used the values that gave the highest performance. For TF-IDF and BOW representations, the thresholds were $K_1 = 0.85$, $K_2 = 0.3$, and $K_3 = 0.07$ for titles and transcripts in Must-Link list and for transcript in Cannot-Link list respectively. For the uSIF representation, the thresholds were $K_1 = 0.85$, $K_2 = 0.65$, and $K_3 = 0$. Having $K_3 = 0$ in uSIF representation does not mean that we exclude the list of Cannot-Link constraint. The cosine similarity values in the uSIF embedding representation can have negative values as some values in the embedding vectors are negatives.

Because PCK-Means and K-Means algorithms produce different clustering assignments for each run (based on how the centroids are initialized), we ran each clustering algorithm 20 times. Then we recorded the average and the max scores. TF-IDF and bag-of-words representations have a total number of 1650 dimension features. So, we reduced the number of dimensions before clustering the data by applying the T-distributed Stochastic Neighbor Embedding (t-SNE) algorithm[12]. We also applied the t-SNE reduction technique on the uSIF embedding. However, because the performance of the uSIF was degraded due to the dimensions reduction, we decided to use all 200 dimensions for the embedding representation: 100 for titles, and 100 for transcripts. Table 2 summarizes the results.

The average and max scores for each algorithm are presented in Table 2. We can see from the table that PCK-Means outperforms K-Means in both representations. The differences in performance between PCK-Means and K-Means are statistically significant, using Welch's t-test, with p-value score < 0.01 in TF-IDF\BOW and

Table 2: The performance of clustering algorithms. PCK-Means outperforms the standard K-Means in both representations: (1) TF-IDF for lecture transcripts and bag-of-words (BOW) for lecture titles, (2) The embedding representation (uSIF) for both lecture transcripts and titles. The performance of PCK-Means is statistically significant (represented by *) in both representations.

Method	AMI		FMI	
	Average	Max	Average	Max
TF-IDF\BOW Representation				
K-Means	0.523	0.597	0.412	0.478
PCK-Means	0.551*	0.649	0.511*	0.632
Embedding Representation (uSIF)				
K-Means	0.395	0.491	0.344	0.452
PCK-Means	0.480*	0.536	0.420*	0.548

uSIF representations for FMI measure. In contrast, when using AMI for the comparison, the differences between PCK-Means and K-Means are statistically significant with p-value scores < 0.01 with uSIF representation and p-value < 0.05 with TF-IDF\BOW representation. We also compare the performance of PCK-Means using different representations: TF-IDF\BOW and uSIF. It is clear from the table that PCK-Means with TF-IDF\BOW representation outperforms PCK-Means with uSIF embedding representation where the difference is statistically significant with p-value < 0.01 in both AMI and FMI measures. In general, PCK-Means with TF-IDF\BOW representation achieves the highest performance.

Since uSIF embedding representation uses pretrained word embeddings that allow it to capture the semantic similarity between documents, we expected it to have the highest performance. However, it did not perform as expected. We investigate this issue and found that some words from our dataset of lecture transcripts and titles do not exist in the list of words from the Glove pretrained model. The total number of missing words was 31 words from both lecture titles and transcripts. The missing words includes some key terms, such as *agglomerative*, *dendrogram*, *subcluster*, *medoids*, *sparkml*, and *dbscan*.

To study the effect of using the lecture titles and transcripts when generating the Must-Link constraint, we compared the performance of the PCK-Means algorithm using only Must-link constraint from titles to the performance of the same algorithm using only Must-link constraint from transcripts. We use TF-IDF and bag-of-words representation with the same set of thresholds for the comparison as PCK-Means achieves the highest performance with this representation. We show the results of this experiment in Table 3. The results indicate that using both lecture titles and transcripts to produce the Must-Link constraint achieves the highest score. We conclude that title and transcripts representations are important for capturing the similarity between lectures. We also notice that removing Must-Link tuples of lecture transcripts reduces the clustering performance more than removing title tuples. This is expected as lecture transcripts contain more keywords than titles. However, using only titles to generate the Must-Link constraint tuples achieves comparable results, which also indicates the importance of using titles to capture lectures similarities.

Table 3: The performance of the clustering in PCK-Means, PCK-No-Title, and PCK-No-Trans using TF-IDF \BOW representation. In PCK-No-Title, we remove all the tuples from Must-Link list that are generated by using lecture titles. In PCK-No-Trans, all Must-Link tuples produced by lecture transcript are removed. Combining both titles and transcripts improves the performance of PCK-Means.

Method	AMI		FMI	
	Average	Max	Average	Max
PCK-Means	0.551	0.649	0.511	0.632
PCK-No-Title	0.534	0.561	0.489	0.576
PCK-No-Trans	0.486	0.559	0.403	0.489

8.3 Examples of Learning Paths

After clustering lectures, we create the precedence graph that represents the six modules by linking the clusters and labeling them with salient terms. We utilize the clusters generated by PCK-Means algorithm with TF-IDF and bag-of-words representation as it achieves the highest performance. In this section, we present several learning paths examples extracted from our precedence graph.

Follow the crowd: The first example we extracted from our precedence graph is the learning path that is shared across many modules in our dataset. Students who follow the most common learning path would have a good overview of the topic as they follow the most popular path that is shared by many courses. Figure 4 depicts the common learning path for the *Unsupervised Learning* topic. The figure shows that the learning path starts with an introduction about Unsupervised Learning, followed by k-Means clustering algorithm and how to choose the number of clusters. Then, dimensionality reduction in clustering is discussed next using the Principle Component Analysis algorithm as an example of dimensionality reduction techniques.

The expert learning path: The second example path we extracted from the precedence graph is one of the longest learning paths. Figure 5 shows a path that spans over seven nodes. This path starts with the partitioning-based clustering methods discussing algorithms, such as “K-Means”, “K-Medians”, “K-Medoids”, and “K-Modes”. It then discusses the application of “K-Means Algorithm” in apache sparkml. Next, it shifts to the hierarchical clustering methods by recommending “Divisive Clustering Algorithm” and “Agglomerative Clustering”. Finally, it presents the “DBSCAN”, a density-based clustering algorithms. This long path is more comprehensive than the common path as it explores more clustering algorithms. Students who are interested in gaining comprehensive knowledge about clustering will find this path very rewarding. Note that such a path does not exist in any of the original six modules we have in our dataset; but it was extracted from the precedence graph constructed by our approach.

Give me some options: Figure 6 shows an example of a sub-graph with several alternative learning paths. To learn the “K-Means” concept, a student can either start with introduction to unsupervised learning or learn about partitioning-based clustering methods. After learning “K-Means”, the student can choose one of the four possible paths: (1) Learn how to choose the number of clusters using “Elbow” methods, (2) Learn about different partitioning algorithms such as “K-Medians”, (3) Move to the hierarchical clustering algorithms and learn “Divisive Algorithm”, or (4) Shift to the hierarchical clustering algorithms and learn “Agglomerative clustering”. Each of these

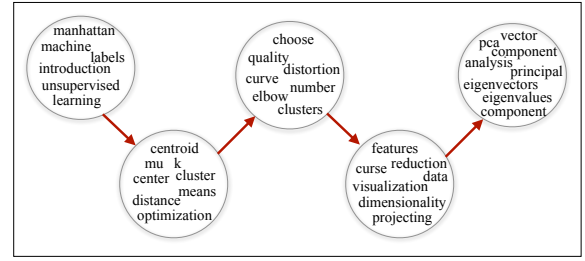


Figure 4: The common learning path extracted from the Precedence Graph. This path is shared by many modules and includes fundamental concepts in Unsupervised Learning topic.

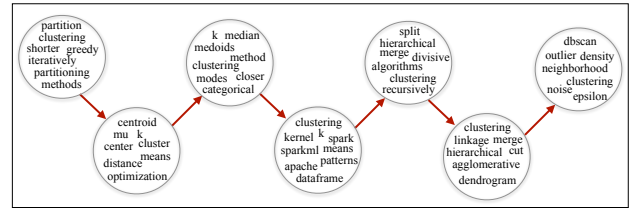


Figure 5: An example of a long learning path extracted from the Precedence Graph. This long path can support students who acquiring expert knowledge as it presents many clustering algorithms.

possible paths are also precedence to other nodes as shown in Figure 6. From the sub-graph, students can choose the learning path that fits their needs. Additionally, the sub-graph shown in Figure 6 gives students a comprehensive overview of how concepts are connected among several courses related to the *Unsupervised Learning* topic.

8.4 Limitations

There are two limitations of our study. First, using the sequence relations among lectures to infer the precedence relations between clusters can cause cycles in the precedence graph. The method proposed in this paper has not addressed the problem of cycles. The naive approach to solve the problem of cycles is to eliminate edges with lower weights that cause cycles in the graph. Further investigation for addressing graph cycles is left as a future work.

Second, in the evaluation we have not examined the performance of our approach in other domains. In the future work, we plan to apply our method on courses from different domain areas and thus generate the precedence graph for each domain.

9. CONCLUSIONS

In this paper, we developed an approach to build the precedence graph of similar MOOCs that have overlaps in content. Our approach is based on Pairwise Constrained K-Means (PCK-Means) clustering algorithm that incorporates constraints to guide the clustering process to focus on clustering similar lectures across courses. We proposed a method of generating the lists of Must-Link and Cannot-Link constraints. PCK-Means with our generated constraint examples significantly outperforms the standard K-Means algorithm with the TF-IDF and bag-of-words representations achieves the highest performance. Using the clusters of similar lectures as nodes in the precedence graph, we connect each cluster according to the

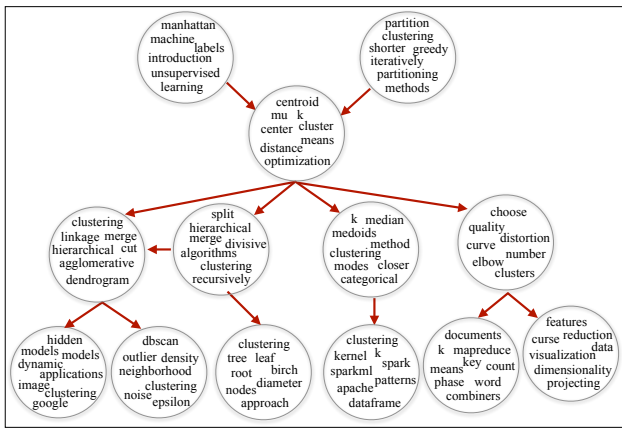


Figure 6: An example of a sub-graph extracted from the Precedence Graph. It depicts some possible paths to learn the topic. It also gives a comprehensive overview about the topic.

precedence relations between lectures in various courses by directed weighted edges to reflect the strength of the precedence relations between clusters. Finally, we label each node in the precedence graph by key concepts extracted from lectures belonging to each cluster. The generated precedence graph reveals popular learning paths as well as alternative learning paths of learning the topics of MOOCs in our dataset.

The precedence graph constructed by our approach is considered the initial block for building applications that support personalized learning. As an example, we can use the precedence graph to build a tool that visualizes the precedence graph to help learners to choose the desired learning paths that are suitable to their interests and backgrounds. Another application is to build a tool that recommends personalized study plans for students based on their interests and time constraints. As discussed in [9, 7], the main obstacle that faces online learners is not having enough time for the course. Further, according to [24, 22], some learners register for a MOOC with a motivation to learn some concepts and hence they drop the course after they are done with studying the concepts of their interest. Wilkowski et al. [22] found that large groups of learners just wanted to learn some concepts without the purpose of earning certificates. Therefore, it is very important to build an application that recommends study plans based on learners motivation, interests, and time constraints. Our proposed precedence graph would be the initial step for building such applications.

10. ACKNOWLEDGMENTS

We would like to thanks all the four participants in the user study for their time and effort to label the dataset.

11. REFERENCES

- [1] R. Agrawal, B. Golshan, and E. Papalexakis. Toward data-driven design of educational courses: A feasibility study. *JEDM- Journal of Educational Data Mining*, 8(1):1–21, 2016.
- [2] F. ALSaad, A. Boughoula, C. Geigle, H. Sundaram, and C. Zhai. Mining mooc lecture transcripts to construct concept dependency graphs. In *Proceedings of the 11th International Conference on Educational Data Mining*, pages 467–473. EDM, 2018.
- [3] S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 333–344. SIAM, 2004.
- [4] O. Borchers. Fast sentence embeddings. https://github.com/oborchers/Fast_Sentence_Embeddings, 2019.
- [5] D. Chaplot and K. R. Koedinger. Data-driven automated induction of prerequisite structure graphs. In *Proceedings of the 9th International Conference on Educational Data Mining*, pages 318–323. EDM, 2016.
- [6] W. Chen, A. S. Lan, D. Cao, C. Brinton, and M. Chiang. Behavioral analysis at scale: Learning course prerequisite structures from learner clickstreams. In *Proceedings of the 11th International Conference on Educational Data Mining*, pages 66–75. EDM, 2018.
- [7] T. Eriksson, T. Adawi, and C. Stöhr. “time is the bottleneck”: a qualitative study exploring why learners drop out of moocs. *Journal of Computing in Higher Education*, 29(1):133–146, 2017.
- [8] K. Ethayarajh. Unsupervised random walk sentence embeddings: A strong but simple baseline. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 91–100, 2018.
- [9] R. F. Kizilcec and S. Halawa. Attrition and achievement gaps in online learning. In *Proceedings of the second (2015) ACM conference on learning@ scale*, pages 57–66. ACM, 2015.
- [10] C. Liang, J. Ye, Z. Wu, B. Pursel, and C. L. Giles. Recovering concept prerequisite relations from university course dependencies. 2017.
- [11] H. Liu, W. Ma, Y. Yang, and J. Carbonell. Learning concept graphs from online educational data. *Journal of Artificial Intelligence Research*, 55:1059–1090, 2016.
- [12] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [13] R. Manrique, J. Sosa, O. Marino, B. P. Nunes, and N. Cardozo. Investigating learning resources precedence relations via concept prerequisite learning. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 198–205. IEEE, 2018.
- [14] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [15] L. Pan, C. Li, J. Li, and J. Tang. Prerequisite relation learning for concepts in moocs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1447–1456, 2017.
- [16] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [17] S. Romano, N. X. Vinh, J. Bailey, and K. Verspoor. Adjusting for chance clustering comparison measures. *The Journal of Machine Learning Research*, 17(1):4635–4666, 2016.
- [18] A. Sanjeev, L. Yingyu, and M. Tengyu. A simple but tough-to-beat baseline for sentence embeddings. *Proceedings of ICLR*, 2017.
- [19] D. Shah. By the numbers: Moocs in 2019 - class central. www.classcentral.com/report/mooc-stats-2019/, 2019.
- [20] S.-s. Shen, H.-y. Lee, S.-w. Li, V. Zue, and L.-s. Lee. Structuring lectures in massive open online courses (moocs) for efficient

- learning by linking similar sections and predicting prerequisites. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
- [21] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al. Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584, 2001.
 - [22] J. Wilkowski, A. Deutsch, and D. M. Russell. Student skill and goal achievement in the mapping with google mooc. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 3–10. ACM, 2014.
 - [23] Y. Yang, H. Liu, J. Carbonell, and W. Ma. Concept graph learning from educational data. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 159–168. ACM, 2015.
 - [24] S. Zheng, M. B. Rosson, P. C. Shih, and J. M. Carroll. Understanding student motivation, behaviors and perceptions in moocs. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, pages 1882–1895. ACM, 2015.

Increasing Enrollment by Optimizing Scholarship Allocations Using Machine Learning and Genetic Algorithms

Lovenoor Aulck
University of Washington
laulck@uw.edu

Dev Nambi
Fred Hutchinson Cancer
Research Center
dnambi@fredhutch.org

Jevin West
University of Washington
jevinw@uw.edu

ABSTRACT

Effectively estimating student enrollment and recruiting students is critical to the success of any university. However, despite having an abundance of data and researchers at the forefront of data science, traditional universities are not fully leveraging machine learning and data mining approaches to improve their enrollment management strategies. In this project, we use data at a large, public university to increase their student enrollment. We do this by first predicting the enrollment of admitted first-year, first-time students using a suite of machine learning classifiers (AUROC = 0.85). We then use the results from these machine learning experiments in conjunction with genetic algorithms to optimize scholarship disbursement. We show the effectiveness of this approach using real-world enrollment metrics. Our optimized model was expected to increase enrollment yield by 15.8% over previous disbursement strategies. After deploying the model and confirming student enrollment decisions, the university actually saw a 23.3% increase in enrollment yield. This resulted in millions of dollars in additional annual tuition revenue and a commitment by the university to employ the method in subsequent enrollment cycles. We see this as a successful case study of how educational institutions can more effectively leverage their data.

Keywords

education, funding, tuition, enrollment management, financial aid

1. INTRODUCTION

Managing student enrollment is one of the core administrative tasks of any university. However, it is far from simple as universities aim to attract and retain the best students with limited resources [4, 10]. Enrollment management has wide-ranging implications on institutions' student body composition as well as their budgeting and finances, where a reliance on tuition income necessitates accurately forecasting

student enrollments [9, 23]. One instrument that has continually been leveraged in the pursuit of enrollments and the associated tuition income is financial aid as receiving a financial aid award increases the likelihood of a student enrolling at the award-giving institution [13, 10]. While financial aid remains a powerful mechanism for institutions to reach their admissions and revenue targets, miscalculating projected student enrollments and mismanaging financial aid funds can have severe implications (such as rescinding over-committed offers¹) [2]. Furthermore, as institutions face tightening budgets and find their pricing policies continually under scrutiny, it remains imperative for them to optimize the resources they have by maximizing enrollments and the associated tuition revenue from financial aid programs [8, 12]. As such, accurately predicting enrollment and optimizing how student aid is disbursed is critical to enrollment management with financial implications that cascade across the entirety of an institution. In this work, we developed an approach to address this challenge, implemented it for a recent entering class, and found that it far outperformed previous strategies.

Predicting enrollment and optimizing the allocation of student aid requires data on student admissions and operational budgets. This data is stored in institutions' organizational databases or can be extracted from operational records. However, despite having this data on previous enrollments and finances, institutions are often slow to leverage it to gain actionable insights and improve institutional processes [20, 26, 14]. What's more, using data for insights in education is less prevalent at traditional campuses (i.e. schools where learning is primarily on-campus) and more common in online and computerized environments, which are more amenable to the collection and analysis of digitized data [17]. To this end, traditional universities remain "data-rich" but are "information-poor" in that they have the raw data needed to extract intelligible insights but are unable to do so due to infrastructure limitations and untrained personnel, among other reasons [21]. This results in the outsourcing of data-centric enrollment work (including developing scholarship disbursement and enrollment strategies) to full-service consulting firms, which do not disclose their proprietary approaches or how their results are evaluated [11]. The lack of motivation for consulting services to disseminate their work coupled with institutions trying to maintain competitive advantages in recruitment limits the extent of

Lovenoor Aulck, Dev Nambi and Jevin West "Increasing Enrollment by Optimizing Scholarship Allocations Using Machine Learning and Genetic Algorithms" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 29 - 38

¹See <https://bit.ly/2Scxqj6> as a recent example.

published research on how institutions can utilize data to improve recruitment processes. As a result, this dearth of literature provides little to demonstrate how data mining and machine learning can assist in the critical mission of enrollment management and in allocating financial aid.

In this project, we mine data from a large, public university in the United States (US) to optimize the disbursement of a merit-based scholarship for domestic non-resident students. We do this in two broad steps. In the first step, we create a predictive model of student enrollment using historical student application data. In the second step, we use a genetic algorithm to optimize scholarship disbursement to maximize student enrollment based on the predictive enrollment model from steps. We conducted this work during a recent admissions cycle of the university and the optimized awards were given to a recent entering class. After seeing improvement in student enrollment yield and an increase of millions of dollars in annual tuition revenue, the university incorporated our approach into their enrollment management process. We believe this project is a case study for other institutions seeking to similarly leverage institutional data for improving enrollment forecasting and financial aid allocation.

2. RELEVANT WORK

The following discussion of relevant work is not exhaustive but is intended to give examples of relevant approaches with a focus on more recent work. While there is some work showing how to predict enrollment, there is very little showing how to allocate scholarships and hardly anything that ties the two together.

2.1 Predicting Enrollment

A few studies have employed machine learning and data mining techniques to predict university enrollment using non-neural approaches. DesJardins developed a logistic regression model using a dataset of approximately 14,400 students from an undisclosed tier I research university in the US. DesJardins' model gave an area under the receiver operating characteristic curve (AUROC) of 0.72 when predicting whether or not a student will enroll [5]. Similarly, Goenner and Paul used logistic regression to predict which of over 15,000 students at a large US university would eventually enroll [7]. Their predictive model gave an AUROC value of 0.87. Nandeshwar and Chaudhari used a suite of learners to predict which of approximately 28,000 students would enroll at West Virginia University [16]. They were interested in variables contributing to students' decisions (finding financial aid to be an important factor) and did not give an assessment of how well their models fared outside of accuracy (which was about 84%).

In addition to the above studies examining non-neural approaches for predicting enrollment, studies have also found that neural approaches fare very well for the same task and often perform better than non-neural approaches. For example, Walczak evaluated different neural network designs when predicting student enrollment at a US liberal arts college, stressing the problem as one of resource allocation [24]. Using a few thousand students, Walczak found that back-propagating neural networks fared best among those compared. Walczak and Sicich later compared neural networks

versus logistic regression to predict enrollment at two US universities [25], finding that neural networks performed better than logistic regression. Chang used logistic regression, decision trees, and neural networks to predict the enrollment of applicants at an undisclosed university, also finding that neural networks outperformed other models when judging by classification accuracy [3].

2.2 Scholarship Optimization

While there are some examples of works examining the use of machine learning in predicting enrollment, there is very little detailing scholarship disbursement strategies, especially ones leveraging machine learning and/or numerical optimization techniques. One example is the work of Alhassan and Lawal, who demonstrated the use of tree-based models for determining which students would be awarded scholarships in Nigeria [1]. Alhassan and Lawal describe the results as "effective" compared to approaches previously used but did not provide additional insight on the success of their approach. Spaulding and Olswang demonstrated the use of discriminant analysis to model the enrollment decisions of students based on varying need-based financial aid awards at an undisclosed university in the US [22]. They found that changes in their award policy would yield only small upticks in enrollment.

One work used machine learning to predict enrollment in conjunction with a numerical optimization technique to disburse scholarships. Sarafraz et al. used neural networks with genetic algorithms to optimize financial aid allocations and while our research is similar in spirit, there are a few notable differences [19]. Firstly, the scholarship fund optimized in this work is merit-based, meaning there are upper and lower bounds on scholarship awards that are specific to each student. This makes for a more difficult optimization task. We also examine alternative predictive models beyond just neural networks (such as ensemble approaches) and use a larger dataset in terms of both the number of observations (i.e. students) and the number of features (over 72,000 observations vs 4,082; over 100 features vs 6). We also provide a comprehensive description of final model performance across multiple metrics and a detailed outline of how genetic algorithms can be used for aid disbursement, including a binning framework to drive the optimization task. Finally, we share real-world enrollment metrics after employing the scholarship optimization to demonstrate the effectiveness of our approach.

3. METHODS

We present the methods for this work by first giving an overview of the setting; then, we describe the data and feature engineering; we then discuss how we predicted enrollment; finally, we discuss optimization constraints and the optimization process. The overall process for this work is shown in Figure 1. Due to the sensitive nature of the data and the fact that it contains personally identifiable information (i.e. student names, addresses, and high schools), we are unable to make it widely available. However, we present the methods below with as much transparency as we can to allow others to replicate the work. We used the Python programming language and implemented feature engineering and predictions using pandas and sci-kit learn, re-

spectively [18]. We developed genetic algorithms using Distributed Evolutionary Algorithms in Python (DEAP) [6].

3.1 Setting

This scholarship optimization work was performed at a large, public US University (the University²). The scholarship fund examined was created to maintain the University’s academic standards while maximizing the enrollment of first-time, first-year (freshmen) domestic non-resident (DNR) students by giving them financial incentive to attend the University. DNR students are students from the US who are not from the state in which the University is located. DNR students account for larger tuition charges than their resident counterparts so their enrollment is of high importance from a budgeting perspective. Tens of millions of dollars in total are awarded annually to these students from the scholarship fund with millions eventually given to students who enroll.

The University is on a quarter-based term system and a vast majority of incoming freshman students start in the fall after applying during the preceding fall and being notified of their acceptance in the preceding spring. The scholarship fund (henceforth referred to as the “DNR scholarships” for domestic non-resident scholarships) was designated to be disbursed for equal amounts across three academic quarters for each of four years (12 quarters total). The DNR scholarships were to be disbursed based on merit. As such, students with higher academic profiles, as defined later, were given equal or larger scholarships than those with lower academic profiles, regardless of financial need. Additionally, only freshmen DNR students who were accepted to the University were eligible for a DNR scholarship award. All admitted DNR students were automatically considered for a DNR scholarship and students did not need to apply for the scholarship.

In years prior, the disbursement strategies for the DNR scholarship were developed by external consulting services. Starting in 2018, the disbursement strategy was brought under the technical stewardship of the University. The first application cycle under the stewardship of the University (i.e. the fall 2018 entering class) is the application cycle for which we optimized scholarship disbursement and detail in this writing. The models that were previously developed for the disbursement of the scholarship fund were proprietary to the consulting services and could not be leveraged. However, student application, enrollment, and scholarship data from prior years was available. When describing results, we compare the results from our approach to that developed by the consulting services. We cannot compare the approach detailed in this writing to a completely un-optimized approach or one that is randomized because the scholarship has never been disbursed in such a manner.

Award-receiving students concurrently learned of the amount of their scholarship and of their admittance to the University. However, not all applications were scored by admissions officers when the first awards were given to students. This was primarily due to the admissions review timeline at the University. As such, we did not know of every admit-

²University administrative offices requested that the institution not be identified.

ted student at the time of optimization yet the scholarship awards were only to be given to admitted students. Thus, the 2018 entering class’s data could not be used directly in the optimizations. Instead, we used data from prior years to develop a fund allocation strategy and then applied this strategy when disbursing scholarships to the 2018 entering class. This was with the expectation that applicants in the 2018 applicant pool were statistically similar to years prior across all the variables used in the modeling and we checked to ensure that this was in fact the case using individual t-tests.

3.2 Data

The data for this work consisted of information on all freshmen DNR applicants to the University from 2014-2017 with usable data. This totaled 72,589 students. The data was compiled from two major institutional sources: the students’ admissions applications and their Free Application for Federal Student Aid (FAFSA) information. The FAFSA is an application prepared by incoming and current US college students to determine their eligibility for financial aid. Examples of data from students’ admissions applications include their high school coursework, entrance exam scores, college GPA (if they had taken classes for credit), whether they were a first-generation college student, and their parents’ educational attainment. These were all self-reported and verified by the University as needed. Data directly from and derived from student FAFSA filings included students’ family income, their expected family contribution to college expenses (as calculated by the University), and loan amounts awarded to the student. About 66% of students had filled a FAFSA. Also included in the data were indicators of whether each student eventually enrolled at the University. Of the 72,589 students in the dataset, 5,081 enrolled (7.00% of all). Demographic variables such as gender and race were available but were not used as discussed in Section 4.1.

The data included tuition amounts students would pay on an annual basis, their financial aid grants and scholarships awarded (outside of DNR scholarship awards), and their DNR scholarship award amount. These variables were not included in any prediction or optimization model on their own. Instead, we created a “`reduced_tuition`” variable which was the annual tuition amount for the students less their total grants and scholarships (i.e. the other two variables summed). We used this variable as a single financial aid and tuition-related feature for the optimization process. This feature is not altered when developing the predictive classifier but is altered during the optimization task, during which the response of students to different award amounts are simulated.

3.3 Feature Engineering

Prior to prediction and optimization, we engineered features from existing variables. First, we either converted categorical variables to dummy variables or replaced them with a binary indicator variable. Then, we grouped students based on their FAFSA award amounts into 6 discrete bins (which were in line with University financial aid record-keeping), each of which was used as a categorical feature. We created binary indications of whether students attended each of the 10 most popular high schools for student applications

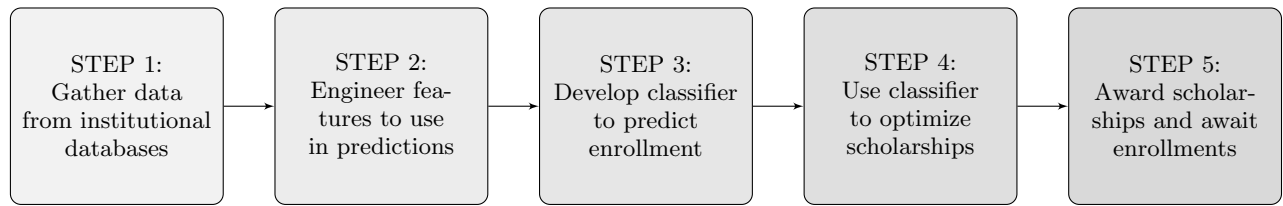


Figure 1: Process for optimizing scholarships, starting with data from institutional databases and ending with disbursements.

and did the same for the 10 most popular states from which students applied. A binary indication was also created for a student athlete designation as each sport had its own application codes. In addition, we also created a separate binary indication for whether the student was transferring any credits from a college in high school program. Students also indicated their academic interests on their applications to the University. We pulled these from their applications and grouped them into 12 broader categories based primarily on the college/department they were associated with at the University (e.g. “Engineering”, “Humanities”, “Health Sciences”, etc.). We then created binary indications of whether a student was interested in each of the categories. Only students’ first application to the University and the resulting admissions/enrollment decisions were included in the data. This ultimately resulted in a total of 108 features extracted from students’ University and FAFSA applications.

Not all applicants filed a FAFSA form and we imputed missing FAFSA-related values. We performed this imputation by building a separate gradient-boosted regression tree model for each FAFSA-related feature using all features that were complete. We then used these regression models to predict the missing values. Only FAFSA-related values were missing and no other features needed to be imputed.

3.4 Predicting Enrollment

To predict enrollment, we first randomly divided the data using a 80-20 training-test split, with 57,359 students in the training set and 14,340 students in the test set. We did not re-balance the data with respect to classes. We scaled the training data by subtracting the median of each feature and dividing by the feature’s interquartile range. We subsequently scaled the test data using the scaling values from the training data. The binary outcome variable indicating whether the student enrolled at the University was not scaled.

After performing the training-test split, we trained 7 machine learning (ML) classifiers on the training set to predict enrollment. These classifiers were: a bagging tree ensemble (BC), gradient boosted trees (XGB), K-nearest neighbors (KNN), random forests (RF), regularized logistic regression (LR), support vector machines (SVM), and a neural network with 3 hidden layers (MLP). We tuned the hyperparameters for each of the classifiers using 5-fold cross validation on the training set. We report performance from all classifiers on the test set, which was not used to train the classifiers and only used to evaluate final performance. We used the classifier with the best performance to optimize scholarship disbursement.

3.5 Optimizing Scholarships

3.5.1 Genetic Algorithms

After developing a classifier to predict enrollment, we used the predictions from the classifier as an objective function in optimization. The aim of the optimization was to develop a strategy that maximized student enrollment from the DNR scholarships. In other words, the optimized approach disbursed scholarships in a manner that maximized the number of students who would enroll at the University from a pool of admitted students to the University. In this work, we used a genetic algorithm (GA) for optimization as GAs are known to work well with a well-defined measure to optimize (i.e. student enrollment) but not a well-defined, continuous, and/or differentiable objective function. GAs are also known to find near-optimal solutions quickly, which was essential when we wanted to rapidly outline and iterate across different budgeting scenarios early in our modeling.

GAs are a class of evolutionary algorithms and are inspired by biological evolution. GAs generally involve iteratively starting with a population of chromosomes, undergoing selection across this population according to a measure of fitness, using genetic crossover and mutation to produce offspring from the most fit individuals, and then using this offspring as the population for the next iteration [15]. The overall population fitness improves with each iteration and the GA eventually converges towards an optimal solution. In this work, we start with a population of award disbursement strategies whose “genetic material” (chromosomes) are a set of scholarship award values; the measure of fitness to assess these individuals is based on predicted enrollment after accounting for constraints; and the crossover and mutation functions used to create offspring are based on altering scholarship award values.

We used the data for the 2017 admitted class in the optimization of scholarship funds. In all, this was 9,479 students (N_{total}). In this sense, we used data from the year prior to optimize the disbursement for the 2018 entering class. We pared the data used in optimizations down to a single year’s application cohort to avoid having to consider if any of the optimization constraints in Section 3.5.3 were being violated for each of the application years simultaneously.

3.5.2 Binning Students

We generated a set of possible scholarship awards that spanned S_{min} to a chosen maximum (S_{max}) in \$300 increments and included \$0. We did not determine S_{max} beforehand but instead set it such that the optimization procedure did not generate an output that included a S_{max} scholarship award. S_{min} was evenly divisible by \$300 and we generated possible scholarship awards in \$300 increments to satisfy constraint

(4) from Section 3.5.3. In all, there were over 20 unique scholarship award values and only these award values were used in the optimizations.

Part of the difficulty of this particular optimization task lies in the fact that awards were to be given in a merit-based manner. As such, the scholarship award for any student is dependent on the awards of students with similar academic profiles. For example, if one was to rank all admitted students in the application pool based on a measure of merit, the minimum possible award given to a particular student would be determined by the award given to the student with the merit that is immediately lower. Similarly, the maximum award for a particular student would be equal to the award given to the student with the merit that is immediately higher. As such, if optimizing on a per-student basis, altering the award for any given student to influence their enrollment decision could result in a cascade that subsequently effects every other student's award amount. This results in a very complex fitness landscape when optimizing scholarship awards on an individual basis.

To resolve this issue of an optimization cascade, we first ranked and then binned students based on academic merit such that all students in the same bin received the same scholarship award. To perform this binning, we sequentially ranked students based on 3 variables: their application academic score, their high school GPA, and their scores on college entrance exams, in that order. This ranking was students' "academic profile." Each student's application academic score was based on a holistic scoring of their academics and was the primary variable for determining their academic profile. We were provided this metric by the University admissions office and it was not calculated/determined by us. Ties between students having the same application academic score were broken by looking at their high school GPA; any remaining ties thereafter were broken using students' entrance exam scores. Once students were ranked, they were divided into 20 ventiles based on their academic profiles (i.e. students were grouped across every 5th percentile) with each ventile receiving the same scholarship award amount. Using ventiles allowed for us to have sufficient flexibility when exploring the fitness landscape during optimization while also not being so granular as to continually be caught in local extrema. Additionally, ventiles helped mitigate the effect of optimization cascades by giving identical awards to students with similar academic profiles. We refer to each of these ventiles as a "bin" and each bin served as the chromosomal building block for the GA. A single scholarship allocation strategy consisted of the scholarship awards across all 20 scholarship bins and is referred to as an "individual" henceforth when used in the context of the GA. Thus, each individual's genetic material can be thought of as being in the form of chromosomes composed of scholarship award bins. It should be noted that we used ventiles after examining the optimization results from other binning strategies (namely using 10, 15, and 25 bins) and finding them to give lower predicted enrollments. We did not, however, attempt to find an optimal bin number beyond this but do intend to explore this in the future.

After binning students, we created a fitness function to evaluate the effect of altering the `reduced_tuition` variable on

student enrollment. Specifically, this function took the genetic material of a scholarship individual (i.e. a set of scholarship awards for each bin) and then re-evaluated the `reduced_tuition` variable for each student based on their updated DNR scholarship award. As noted above, we created the `reduced_tuition` variable by taking the tuition due for a student and subtracting their total grants and scholarships; it was the only financial aid and tuition-related variable used in the predictive model. The function re-calculated each student's likelihood for enrollment based on the updated values for `reduced_tuition` using the predictive enrollment model. The final output for the fitness function was a calculation of the number of students predicted to enroll for a given scholarship individual, which we used as the fitness criterion for evaluating individuals.

3.5.3 Modeling Constraints

Several constraints were posed on the scholarship disbursement by University administrators. Due to University policy, exact values for awards and budgets will not be discussed. Some constraints on the disbursement strategy were as follow, where F represents funds in DNR scholarship offers, B represents funds in the DNR scholarship budget, N specifies a count of students, and S specifies a scholarship award amount:

1. The total amount spent on DNR scholarships (F_{spent}) cannot exceed a pre-determined amount (B_{spent}):

$$F_{\text{spent}} \leq B_{\text{spent}}$$
2. The total amount offered to students in DNR scholarships regardless of whether they enroll (F_{offered}) cannot exceed a pre-determined amount (B_{offered}):

$$F_{\text{offered}} \leq B_{\text{offered}}$$
3. The percentage of admitted students who are awarded scholarships ($N_{\% \text{awarded}}$) should be approximately equal to a pre-determined percentage ($N_{\% \text{target}}$):

$$N_{\% \text{awarded}} \approx N_{\% \text{target}}$$
4. The award amounts must be divisible by \$300 to allow for round hundred-dollar splits across three academic terms.
5. There is a minimum value for a single scholarship award (S_{min}) but no pre-determined maximum value.

The organization of the population, individuals, and bins for the GA optimization is shown in Figure 2. We generated an initial population of p individuals by randomly selecting K scholarship awards (one for each bin) from the set of possible scholarship awards and sorting for each individual. For this work, $p = 1000$ and $K = 20$. Each bin contained the same number of students (N_{bin}), which was equal to $\frac{N_{\text{total}}}{K}$. All students in the same bin received the same award for a given individual; awards were not unique to each bin and could be duplicated across a given individual. N_{bin} multiplied by the scholarship award value for each bin equalled the funds awarded for that respective bin; the sum of these across all K scholarship bins for an individual was F_{offered} for that individual. The predicted number of enrollees for each scholarship bin multiplied by the award for that respective bin equalled the funds spent for that bin; the sum

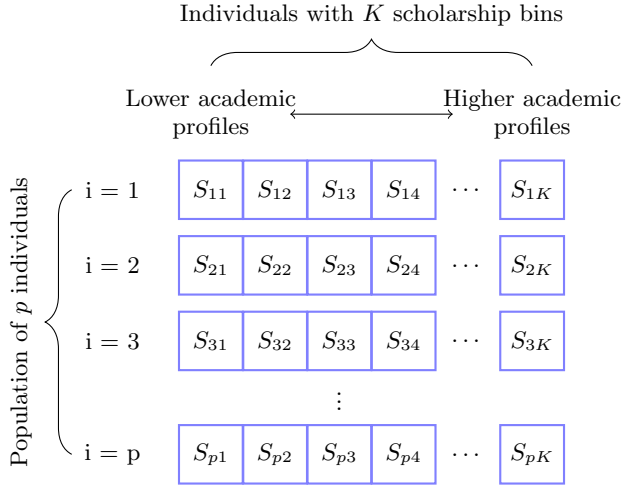


Figure 2: Genetic algorithm setup. Individuals (i) are scholarship allocation strategies of K scholarship bins (j). The population consists of p individuals. Each S_{ij} is a scholarship award value for the i^{th} individual and the j^{th} scholarship bin. The bins are sorted based on academic profile such that $S_{i1} \leq S_{i2} \leq S_{i3} \dots \leq S_{iK}$ for any given i (but not necessarily across individuals). For this work, $K = 20$ and $p = 1000$.

of these across all K scholarship bins for an individual was F_{spent} for that individual. The number of bins with non-zero award values divided by K was equal to $N_{\% \text{awarded}}$ for an individual.

We penalized each individual's fitness if the optimization constraints above were violated. We initialized a single penalty coefficient (σ) to 1.0 and then successively enforced each of the following squared penalties for a given scholarship individual:

- if too much was spent on scholarship awards:
 $F_{\text{spent}} > B_{\text{spent}} \rightarrow \sigma = \sigma * \left(\frac{B_{\text{spent}}}{F_{\text{spent}}}\right)^2$
- if too much was offered in scholarship awards:
 $F_{\text{offered}} > B_{\text{offered}} \rightarrow \sigma = \sigma * \left(\frac{B_{\text{offered}}}{F_{\text{offered}}}\right)^2$
- if too many students were awarded a scholarship:
 $N_{\% \text{awarded}} > N_{\% \text{target}} \rightarrow \sigma = \sigma * \left(\frac{N_{\% \text{target}}}{N_{\% \text{awarded}}}\right)^2$
- if too few students were awarded a scholarship:
 $N_{\% \text{awarded}} < N_{\% \text{target}} \rightarrow \sigma = \sigma * \left(\frac{N_{\% \text{awarded}}}{N_{\% \text{target}}}\right)^2$

Ultimately, we multiplied the output of the fitness function (i.e. the predicted enrollment count for an individual) by the penalty coefficient to penalize constraint-violating individuals. If there were no constraints violated, the penalty coefficient was 1.0 and the fitness evaluation of the individual remained unchanged.

3.5.4 Optimization Process

The approach for the GA was as follows. We randomly generated the initial population of individuals as described

above. We then calculated the fitness of each individual and took a subset of the most fit individuals (10%) as the basis for the next generation of the population. We then employed genetic crossover to this subset of the population to generate offspring. We used two-point genetic crossover, wherein two points were randomly selected along chromosomes and the genetic material from one individual was swapped with that from another between the two points, much like a two-point crossover mutation in nature. In other words, for a pair of randomly selected individuals, we randomly selected two scholarship bins from ventiles 1 through 20 and all scholarship award values between the two bins from one individual were swapped with those from the other individual and vice versa.

After using crossover to refill the population, the offspring underwent mutation. We used three types of mutations: an increase mutation, a decrease mutation, and a swap mutation. For a mutation, we randomly selected an individual and then randomly selected a bin from this individual. The award for this bin was either increased to another possible award amount (increase mutation), decreased to another possible award amount (decrease mutation), or swapped for another randomly selected award amount (swap mutation). The probability of performing either an increase, decrease, or swap mutation were equal unless the scholarship award value equaled S_{\min} or S_{\max} , in which case we eliminated the possibility of a decrease or an increase mutation, respectively. Once a particular mutation was selected for a given individual and bin, a single award value was randomly selected from all possible award values that satisfied the condition of the mutation and used in the mutation. After mutations, we re-sorted the awards across each individual to ensure students with higher academic profiles received larger awards. We kept the initial subset of the most fit individuals unchanged during crossover and mutation; instead, we altered replicas of these individuals to compare the most fit individuals from one generation to those from the next generation. The new generation of individuals then served as the population for the next algorithmic iteration. We repeated this process for 20 generations of the population and used the most fit individual thereafter as the scholarship allocation strategy. The process for the GA is shown in Process 1.

Process 1: Genetic algorithm process for scholarship allocation (parameters for this work are in parentheses)

- 1: Initialize population ($p = 1000$ with $K = 20$ bins each)
- 2: Evaluate fitness of each individual (where fitness is enrollment count predicted by classifier)
- 3: For each of G generations: ($G = 20$)
- 4: Keep subset of population with highest fitness (10%)
- 5: Use two-point crossover to fill population
- 6: Mutate random bins of random individuals
- 7: Evaluate fitness of each individual
- 8: Use individual with highest fitness after G generations

4. RESULTS AND DISCUSSION

Using the methods described in Section 3, we developed a predictive classifier of student enrollment and used it in conjunction with a genetic algorithm that optimized the allocation of a scholarship fund. Ultimately, the university saw a

Table 1: Classifier performance sorted by rank across all metrics. Names of classifiers are provided in Section 3.4.

	Model	Accuracy	AUC	F1-score
1.	XGB	93.10%	0.846	0.905
2.	RF	93.06%	0.848	0.901
3.	MLP	93.01%	0.845	0.902
4.	BC	93.05%	0.833	0.901
5.	LR	92.96%	0.805	0.900
6.	SVM	93.00%	0.780	0.900
7.	KNN	92.80%	0.793	0.893

23.8% increase in enrollment yield after using our approach. This resulted in millions of dollars of additional annual tuition revenue. The following section presents these results in greater detail in the same order as the methods.

4.1 Predicting Enrollment

Previous studies have shown the effectiveness of ML in predicting enrollment. We examined seven different predictive classifiers for this task. We show the performance of these classifiers in terms of prediction accuracy, AUROC, and F1-score in Table 1. We used the same observations as a test set when comparing performance across classifiers; for the test set, the majority class represented 92.8% of observations (i.e. 7.2% of students in the test set eventually enrolled at the University). All classifiers performed similarly in terms of both accuracy and F1-score. Because of the large class imbalance, there were only modest gains in terms of accuracy over the majority class representation. Ensemble classifiers (RF, XGB, and BC) had the highest accuracies while KNN performed on par with the majority class representation (note: it was checked that the KNN model did not predict that all observations were of the majority class). The highest F1-score, meanwhile, was given by the XGB classifier, though it was not substantially higher than other classifiers.

We show ROC curves for the classifiers in Figure 3. The general shape of the ROC curves was similar across the classifiers but with meaningful variation in AUROC. Specifically, RF, XGB, and MLP tended to perform similarly in terms of AUROC and had the highest AUROC values. This is in line with previous work where neural networks tended to perform well when predicting enrollment, even without more complex architectures in this case. That said, the ensemble classifiers performed similarly well for the task at hand.

Demographic data was not used in the models. Including demographic variables in the prediction models would improve predictive performance to some degree, albeit at the expense of potential explicit discrimination with respect to recipient characteristics. As such, we decided to exclude demographic variables when building the classifiers. While doing so limits the degree of explicit discrimination, the possibility of implicit discrimination remains - particularly with respect to associations between demographics, income, geography, and academics. Checking and controlling potential demographic imbalances is beyond the scope of this particular

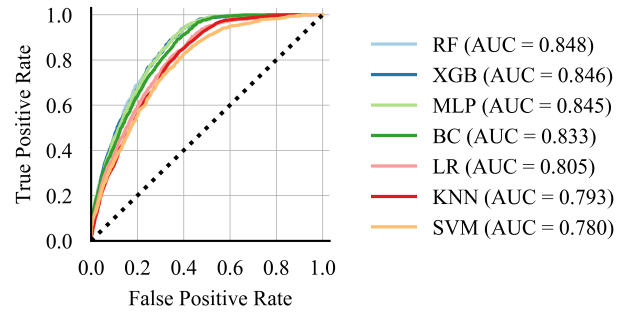


Figure 3: ROC curves for enrollment prediction

work but was handled by stewards of the DNR scholarship fund after optimization. It should again be noted that the DNR scholarships were designated to be awarded in a merit-based manner and financial need was not be considered in the allocation process.

We examined classifier performance across all metrics and decided to use XGB when optimizing scholarship allocation. Prior to optimization, we calibrated the classification threshold for the prediction probability to the nearest one-hundredth such that the number of students predicted to enroll by the model was nearest to the actual enrollment count. By calibrating the threshold in this manner, we used a lower probability decision threshold (0.22) than the value of 0.5 that is typically used in binary classification. We understood that doing so came at the expense of an increased rate of false positives (Type I error) but it also allowed for the predicted enrollment counts to be closer to actual counts, which was necessary when discussing predictions with administrative stakeholders. We show the effects of this calibration in Figure 4, where the confusion matrix using the standard classification threshold of 0.5 is shown along with the confusion matrix using the calibrated threshold of 0.22.

Of note from the confusion matrices is how well students who did not enroll at the University could be identified. On the other hand, it was much more challenging to identify those who would enroll. This speaks to the selectivity of the University in that many of the candidates who would not enroll were simply those who were not accepted to the University (students' acceptance to the University was not included as

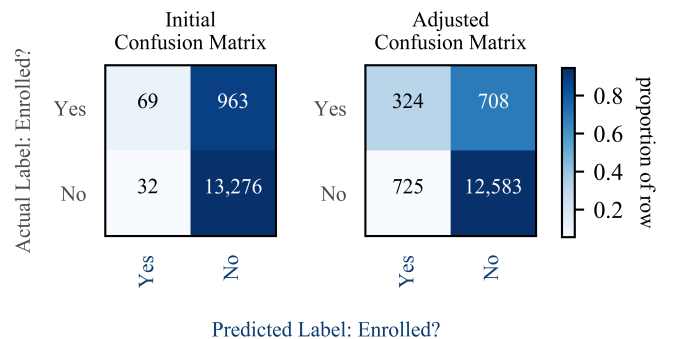


Figure 4: Confusion matrices for predicting enrollment using XGB and a classification threshold of 0.5 (left) and a calibrated classification threshold of 0.22 (right)

Table 2: Predicted enrollments after calibrating the classification threshold for test data and all data (training + test data).

	Test Data	All Data
Actual	1,032	5,081
Predicted	1,049	5,166

a feature during predictions). Concurrently, the difficulty with identifying students who will enroll aligns with the fact that these DNR students are applying to a university that is away from their respective homes and social bases. Also, those that are accepted to the University tend to be of higher academic standing, giving them more potential college choices. Thus, the general likelihood of a DNR student enrolling is difficult to determine when considering potential social factors and college options.

Lowering the classification threshold resulted in predicted enrollment counts in line with what was seen in the data, as shown in Table 2. Calibrating the classification threshold also allowed for a greater number of true positives while also balancing the number of false positives and false negatives. We also examined the effect of similarly calibrating the classification thresholds when using the other ML classifiers and determined that using XGB would still be viable for scholarship optimization.

4.2 Optimizing Scholarships

After we developed a model for predicting student enrollment, we used a GA to design a scholarship disbursement strategy. We used the GA in a setup with students grouped in ventiles and each ventile receiving the same award amount. The genetic material (awards for each ventile) for individuals (allocation strategies) was altered for each iteration of the GA and then fitness was determined. Fitness was based on predicted enrollment after accounting for the violation of constraints. Due to the application review timeline at the University, we did not know which students of the 2018 entering class would be admitted and used the prior year’s application data (2017) to develop a disbursement strategy. Because the disbursement strategy relied on students being grouped into ventiles, we applied it to the most recent entering class after checking that the two classes were similar across academic-related variables using paired t-tests and chi-squared tests. Additionally, the binning strategy and the use of ventiles alleviated concerns about the size of the entering class as specific award amounts were disbursed to proportions of the admitted class and not to a fixed count thereof.

We show fitness (predicted student enrollment) measures across the population of individuals for each generation of the GA in Figure 5. As expected, the maximum, mean, and median values of fitness increase across generations, though these increases are much smaller for later generations. The minimum fitness values for the population follow a similar trend with some variation. All metrics eventually converge to the predicted enrollment, which is shown as a percentage. We intend to use Monte Carlo simulations in the future to outline a distribution of likely enrollment counts during the

optimization process.

The exact award amounts for the DNR scholarship cannot be disclosed due to University policy. Additionally, the percentage of students receiving scholarship awards was not consistent across previous years. For example, in some years, 30% of accepted DNR students may receive a scholarship while in other years, 70% of accepted DNR students may receive a scholarship. Furthermore, tuition charges change annually at the University. Thus, in an attempt to provide a normalized measure for comparison across entering classes without disclosing exact award amounts, we compare award allocation strategies across time based on the discount on tuition. For example, a student receiving a \$5,000 scholarship when tuition is \$20,000 receives a 25% discount on tuition. We show previous allocations of the DNR scholarship to scholarship-receiving students as a discount on tuition in Figure 6. This discount on tuition factors in tuition cost for a full-time DNR student but not additional living or educational expenses (i.e. housing, food, books, etc). To further illustrate the use of discount on tuition, when looking at Figure 6, it can be seen that approximately 15% of all scholarship-receiving students received an award that discounted their tuition by 8-12% in 2014 while in 2017, approximately 60% of students received a similar award. For each of the bands in Figure 6 (six bands per entering class), only a single scholarship award amount fitting within a given band was given to students for a single entering class. It is apparent from examining previous allocations that the manner in which the awards were historically allocated shifted greatly from year to year. As noted previously, these previous allocations were determined by a external consulting services and we could not leverage their underlying approach or insight in this work.

We also show the scholarship allocation strategy for the 2018 entering class (for which the scholarship disbursement was optimized in this project) in Figure 6. This strategy tended to favor smaller scholarships, which aligns with the optimized allocation strategy that Sarafraz et al. reported [19]. In fact, scholarship stewards had initially placed a lower limit on the scholarship awards (S_{\min}) during modeling, which was equal to the lowest scholarship amount that had historically been awarded to students. This lower limit was between a 8-12% discount on tuition. After we discussed preliminary results of the optimization and the effectiveness of smaller awards with the scholarship stewards, it was determined that the lower limit on the awards would be changed

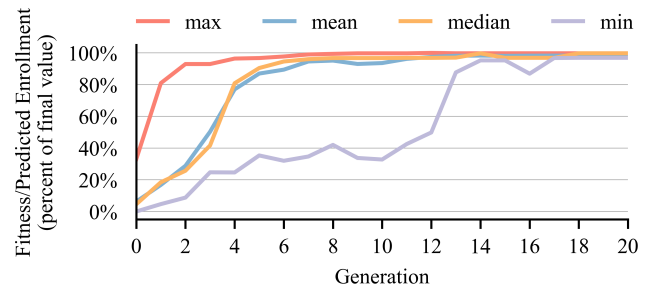


Figure 5: Fitness measures across generations of genetic algorithm. Fitness was equivalent to predicted enrollment.

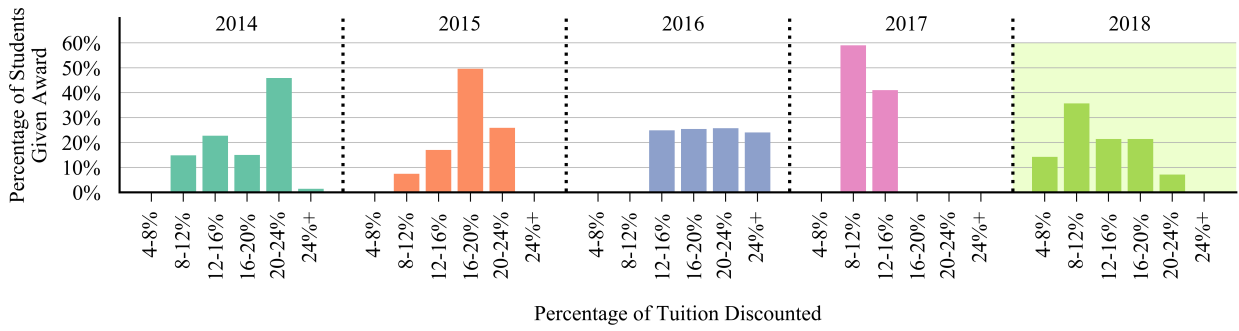


Figure 6: Historical scholarship allocations for the DNR scholarship. The highlighted year (2018) shows the optimized scholarship allocations from this work. Upper bounds for the bins indicating discounted tuition are inclusive. Percentages of students given awards are percentages of award-receiving students only.

Table 3: Historical, predicted, and actual yields after scholarship disbursement.

	Timeframe	Yield	% Increase
Historical	2014-2017	10-12%	N/A
Predicted	2018	13.9%	15.8%
Actual	2018	14.8%	23.3%

to $\frac{S_{\min}}{2}$. Thus, the 2018 entering class had some scholarship awards that were lower than those received by previous entering classes. These lower awards discounted tuition by 4-8%. It is also noteworthy that the optimized disbursement strategy gave a distribution of awards that was right-skewed (with more of the awards being lower in value), in contrast to previous allocation strategies, which were predominantly left-skewed (with more of the awards being higher in value) or near uniform. This speaks to the idea that smaller scholarships awarded to students of lower merit may be more effective than larger scholarships are for those of higher merit (keeping in mind that students who received smaller awards were also of lower merit for this merit-based scholarship). This aligns with intuition that those with higher academic profiles have more college options and require additional recruitment, be it additional financial aid or in some other form. It could also relate to the idea that higher-performing students come from more advantaged socioeconomic backgrounds, thereby diminishing the effect a scholarship may have on their enrollment decisions.

After we developed the scholarship distribution strategy for the 2018 entering class, the University distributed scholarship awards to admitted DNR freshmen. We then waited as these students indicated their enrollment decisions a few months later. In recent years, the yield for DNR students at the University was about 10-12% with little/no increase, as verified by scholarship stewards, where “yield” refers to the percentage of admitted students who enrolled at the University. Historical yields were not based on an un-optimized or randomized scholarship allocation strategy but were the product of the scholarship allocations derived by external consulting services. Thus, because we were comparing the results from our approach to those from a previously optimized strategy (and not an un-optimized or random allocation strategy), we expected to see a modest improvement,

if any at all. Instead, we saw a much higher increase in yield. Table 3 shows the historical yields, the predicted yield based on our optimized approach, and the actual yield based on student enrollment for the 2018 entering class. When comparing to the upper bound on historical yield (12%), we anticipated that the scholarship optimizations would increase student yield by 15.8% (12% to 13.9%) based on the enrollment numbers we had seen during the optimizations (which was computed using XGB and the calibrated classification threshold). In reality, yield increased by 23.3%. This amounted to hundreds of additional students enrolling with each paying tens of thousands of dollars annually in tuition. There was also no discernible difference between the academic aptitude of students from the 2018 entering class and years prior. Overall, the net effect was an increase in millions of dollars in annual tuition revenue for the University. The University has since incorporated our approach into their enrollment modeling process for future disbursements of this scholarship fund. Of note is that yields are based on proportions of students that enrolled and the size of the entering class makes little difference when comparing yields. The University also admitted roughly the same percentage of DNR students as years past and nearly all conditions during the application process were identical to previous entering classes. That said, the degree to which this increased yield can be causally attributed to the scholarship optimizations warrants further investigation. This may be in the form of A/B testing or some other controlled experiment.

5. CONCLUSIONS

In this work, we show how existing data at a university can be used to improve enrollment management. We combine machine learning with numerical optimization and use student application data at a public university to optimize a scholarship fund. We find that the optimized approach increased student enrollment and generated millions in tuition revenue. Our approach has been incorporated into the university’s enrollment forecasting.

We show that ensemble classifiers can give strong performance when predicting enrollment and we use a binning strategy based on student merit to make the optimization task more tractable. This strategy eliminated the need for per-student optimizations, thereby limiting the complexity of the fitness landscape during optimization. After optimization,

tion, we see that smaller scholarship awards work better for maximizing enrollment. In all, the University had historically seen little/no increase in enrollment yield and we projected that our optimized approach would increase yield by 15.8%. In reality, enrollment yield increased by 23.3%.

Universities are at the forefront of training the next generation of data scientists and developing data-centric tools and techniques. However, they are far behind in applying data science to their own administrative data and processes. This project attempted to move them in this direction. Using a suite of machine learning tools, we were able to increase a university's revenue from a scholarship fund by millions of dollars. We think there are many similar opportunities to harness the power of data science in the realm of education administration, especially in resource allocation.

6. REFERENCES

- [1] J. Alhassan and S. Lawal. Using data mining technique for scholarship disbursement. *World Academy of Science, Engineering and Technology, International Journal of Computer and Information Engineering*, 2(7), 2015.
- [2] C. M. Antons and E. N. Maltz. Expanding the role of institutional research at small private universities: A case study in enrollment management using data mining. *New directions for institutional research*, 2006(131):69–81, 2006.
- [3] L. Chang. Applying data mining to predict college admissions yield: A case study. *New Directions for Institutional Research*, 2006(131):53–68, 2006.
- [4] M. D. Coomes. The historical roots of enrollment management. *New directions for student services*, 2000(89):5–18, 2000.
- [5] S. L. DesJardins. An analytic strategy to assist institutional recruitment and marketing efforts. *Research in Higher education*, 43(5):531–553, 2002.
- [6] F.-A. Fortin, F.-M. D. Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné. Deap: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13(Jul):2171–2175, 2012.
- [7] C. F. Goenner and K. Pauls. A predictive model of inquiry to enrollment. *Research in Higher education*, 47(8):935–956, 2006.
- [8] J. Hood. The new austerity: University budgets in the 1990s. *Academic Questions*, 9(2):82–88, 1996.
- [9] D. S. Hopkins. *Planning models for colleges and universities*. Stanford University Press, 1981.
- [10] D. Hossler. The role of financial aid in enrollment management. *New directions for student services*, 2000(89):77–90, 2000.
- [11] D. Hossler. Enrollment management & the enrollment industry. *College and University*, 85(2):2, 2009.
- [12] H. A. Hovey. State spending for higher education in the next decade: The battle to sustain current support. 1999.
- [13] L. L. Leslie and P. T. Brinkman. Student price response in higher education: The student demand studies. *The Journal of Higher Education*, 58(2):181–204, 1987.
- [14] T. E. D. Mining. Enhancing teaching and learning through educational data mining and learning analytics: An issue brief. In *Proceedings of conference on advanced technology for education*, 2012.
- [15] M. Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.
- [16] A. Nandeshwar and S. Chaudhari. Enrollment prediction models using data mining. Retrieved January, 10:2010, 2009.
- [17] D. Niemi and E. Gitin. Using big data to predict student dropouts: Technology affordances for research. In *Proceedings of the International Association for Development of the Information Society (IADIS) International Conference on Cognition and Exploratory Learning in Digital Age*, 2012.
- [18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.
- [19] Z. Sarafray, H. Sarafray, M. Sayeh, and J. Nicklow. Student yield maximization using genetic algorithm on a predictive enrollment neural network model. *Procedia Computer Science*, 61:341–348, 2015.
- [20] X. Shacklock. *From bricks to clicks: the potential of data and analytics in higher education*. Higher Education Commission London, 2016.
- [21] F. Siraj and M. A. Abdoulha. Uncovering hidden information within university's student enrollment data using data mining. In *Modelling & Simulation, 2009. AMS'09. Third Asia International Conference on*, pages 413–418. IEEE, 2009.
- [22] R. Spaulding and S. Olswang. Maximizing enrollment yield through financial aid packaging policies. *Journal of Student Financial Aid*, 35(1):3, 2005.
- [23] D. Trusheim and C. Rylee. Predictive modeling: linking enrollment and budgeting. *Planning for Higher Education*, 40(1):12, 2011.
- [24] S. Walczak. Neural network models for a resource allocation problem. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(2):276–284, 1998.
- [25] S. Walczak and T. Sincich. A comparative analysis of regression and neural networks for university admissions. *Information Sciences*, 119(1-2):1–20, 1999.
- [26] D. M. West. Big data for education: Data mining, data analytics, and web dashboards. 2012.

“Hello, [REDACTED]”: Protecting Student Privacy in Analyses of Online Discussion Forums

Nigel Bosch
University of Illinois at Urbana–
Champaign
Champaign, IL, USA
pnb@illinois.edu

R. Wes Crues
University of Illinois at Urbana–
Champaign
Champaign, IL, USA
rwcrues@gmail.com

Najmuddin Shaik
University of Illinois at Urbana–
Champaign
Champaign, IL, USA
shaik@illinois.edu

Luc Paquette
University of Illinois at Urbana–
Champaign
Champaign, IL, USA
lpaq@illinois.edu

ABSTRACT

Online courses often include discussion forums, which provide a rich source of data to better understand and improve students’ learning experiences. However, forum messages frequently contain private information that prevents researchers from analyzing these data. We present a method for discovering and redacting private information including names, nicknames, employers, hometowns, and contact information. The method utilizes set operations to restrict the list of words that might be private information, which are then confirmed as private or not private via manual annotation or machine learning. To test the method, two raters manually annotated a corpus of words from an online course’s discussion forum. We then trained an ensemble machine learning model to automate the annotation task, achieving 95.4% recall and .979 AUC (area under the receiver operating characteristic curve) on a held-out dataset obtained from the same course offered 2 years later, and 97.0% recall and .956 AUC on a held-out dataset from a different online course. This work was motivated by research questions about students’ interactions with online courses that proved unanswerable without access to anonymized forum data, which we discuss. Finally, we queried two online course instructors about their perspectives on this work, and provide their perspectives on additional potential applications.

Keywords

Text anonymization; discussion forums; online learning.

1. INTRODUCTION

Online education is an essential part of many university programs [12] and has many potential benefits, such as convenience, scalability, and lower cost for both students and institutions. However, personal connections and discussions with fellow students could be quite negatively impacted if there are no

opportunities for students to interact with each other as they can easily do in face-to-face classes. Hence, many online courses include optional or required discussion forums, in which students can talk about course content or connect with each other. For researchers, the textual contents of these forums is a valuable source of knowledge for understanding more deeply how students experience learning in online environments (see studies such as [4, 6, 8, 11, 16, 18, 23, 26, 37]). A significant barrier to analyzing the contents of these forums is the private nature of information students can and do disclose to each other, such as names, affiliations, locations, and contact information. Analyzing these data often requires anonymization before researchers can ethically and legally access the data for analyses. In this paper, we propose and evaluate a method specifically designed for anonymizing student-generated text in discussion forums.

There are various types of identifying information students share on discussion forums. Some are relatively straightforward to remove, such as phone numbers and email addresses, which follow a relatively limited set of formatting patterns. Others are less predictable – especially the names of people and places, which can appear in various forms (e.g., nicknames), overlap with dictionary words (e.g., May, Lane, Bob), or refer to entities not listed in course rosters (e.g., family members, locations). For example, one student in data we analyzed posted potentially identifying information about a pet:

“Hello [REDACTED], I am also a pet lover. I have a [REDACTED] schnauzer, whose name is [REDACTED]. What’s your work at the dog kennel? How many puppies are there in the kennel? It seems lots of fun!”

While other students refer to themselves or others by alternate names, as in the case of this student:

“Hi guys, My name is [REDACTED], but I prefer to be called [REDACTED]. I was born and grew up in [REDACTED], but I moved to [REDACTED] when I was in 7th grade.”

Moreover, students frequently misspell identifying and non-identifying information (e.g., “*Battlestar Gallactica*”, “*When we icnrease entropy does it change delta G as well?*”), which – combined with grammatical errors – resulted in relatively poor anonymization quality in our early efforts built on named entity

Nigel Bosch, Wes Crues, Najmuddin Shaik and Luc Paquette
“Hello, [REDACTED]”: Protecting Student Privacy in Analyses of Online Discussion Forums” In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 39 - 49

recognition software. Hence, we sought more robust methods to detect identifying information to be redacted.

Identifying information can occur in forums when students organize study groups, address questions and answers to each other, and other situations. Students may receive meaningful benefits from exposing private information online – for example, if it enables them to connect more closely with peers they may never meet offline. Examples such as those above are especially common in introductory discussion forums at the beginning of courses, where students get to know each other. However, the presence of identifying information prevents researchers at many academic institutions offering online courses from analyzing forum data (at least without individual permission from each student), and thus from enhancing student learning experiences through applications of research. We focus on this problem for the specific case of university-level online courses, of which there are many, and propose an automated text anonymization solution that rivals human accuracy, despite the variance in form, content, and spelling inherent in student-generated text.

1.1 Privacy Concepts and Anonymization Strategies for Text

There is a large body of previous research on removing identifying information from text. A primary focus of prior work has been specifically on removing names and identifying information from medical records (see [24] for a review). One of the earliest methods employed a template-matching approach to find names, addresses, phone numbers, and other identifying information in medical records (e.g., notes written by doctors) [35]. Later research with similar methods has shown that template-matching approaches can be quite accurate in held-out (unseen) medical records data, achieving a recall of .943 [28], which compared favorably to inter-human agreement on the same data.

Early work on anonymizing text also led to the concept of k -anonymity [33, 34], in which a formal guarantee is made for a particular dataset that every person in that dataset is indistinguishable from $k-1$ or more other people in the dataset. This has resulted in additional text anonymization research that goes beyond names of people and places to include identifying characteristics such as specific diseases and treatments that may be sufficiently unique to reduce k with some effort [3]. In general, these works utilize lists of known names and forms of names (e.g., “Dr. [name]”) to identify words for removal in text – forms which are used infrequently in online course discussion forums – and tend to focus on the unique needs of medical literature anonymization.

Named entity recognition (NER) is another closely-related field that focuses on finding *and classifying* names in text [25]. Modern NER approaches typically rely on machine learning to discover names in text by learning from large corpora of annotated or partially-annotated text. In theory, NER can be applied for anonymization purposes by finding names and removing or replacing those from classified categories of interest (i.e., people, places, and organizations that may be employers) [10]. However, modern NER systems are typically trained on large amounts of data that differs considerably from discussion forum data (e.g., the entire contents of Wikipedia), and do not generalize well to new domains [20, 21].

Previous research has also studied privacy and anonymity in structured data (e.g., directed graphs, tabular records) that is relevant to forum anonymization. For example, social network analysis shows that individuals in one social network can be

identified in another network based on who they interact with [27], which might occur across course discussion forums. The network of semantic and stylistic relationships between words can also identify individuals from text data [2, 5]. Such connections have led to the concept of differential privacy. Differential privacy is one of the strongest types of data privacy [14], which guarantees that it is impossible to determine whether or not a query individual’s data was included in a given dataset or result. While we do not propose providing such a strong guarantee for anonymizing discussion forum text for research analyses – given the need for obfuscating much of the text that could be needed for analyses (e.g., person-specific sentiment words) – we instead propose a set of goals that allow well-intentioned researchers to access data with minimal exposure to identifying information.

1.2 Novelty of the Problem

Our method for automatically anonymizing discussion forum text aims to satisfy several goals needed for practical application. Specifically, the automatic method should:

- 1) Achieve accuracy similar to human accuracy, if it is to be used as a replacement for manual annotation
- 2) Not require annotation of large amounts of domain-specific text data for development or validation
- 3) Not rely on lists of student names, which may be unavailable (as was the case in our work), may not capture the diversity in naming conventions of students from various cultures, and may not capture nicknames frequently used by students

Approaches relying on NER methods satisfy goals 1 and 3 well, but not goal 2. Conversely, approaches developed for the needs of medical record anonymization typically satisfy goal 1 and potentially goal 2 (though this has not been well tested and may not be the case if the style of medical text differs notably from online discussion forum text), but typically do require information such as lists of individual’s names, and do not satisfy goal 3.

The approach we propose here satisfies the three goals outlined above, utilizing a set-theoretic approach to drastically reduce the burden of manual annotation and machine learning to further automate the manual annotation process.

2. ANONYMIZATION METHOD

The anonymization procedure consists of three broad steps (see Figure 1 for an overview). First, we extract a set of possible name words from the discussion forum text. Second, we classify possible names as either actual names or not names, via manual annotation or machine learning. Third, we remove the identified names from text, along with other likely identifying information that can be found via regular expressions, including emails, URLs, and phone numbers.

2.1 Data Collection

We obtained discussion forum text data from two online courses offered at a large, public university in the Midwestern United States. The first course (*course 1*) was an elective STEM course offered using the Moodle learning management system [13], while the second course (*course 2*) was an introductory STEM course that was required for students in some majors and was offered using the LON-CAPA learning management system [22]. Discussion forum participation was a required, graded component of both courses, and students were quite active in the forums. We obtained two semesters of course 1 data separated by two years and one semester

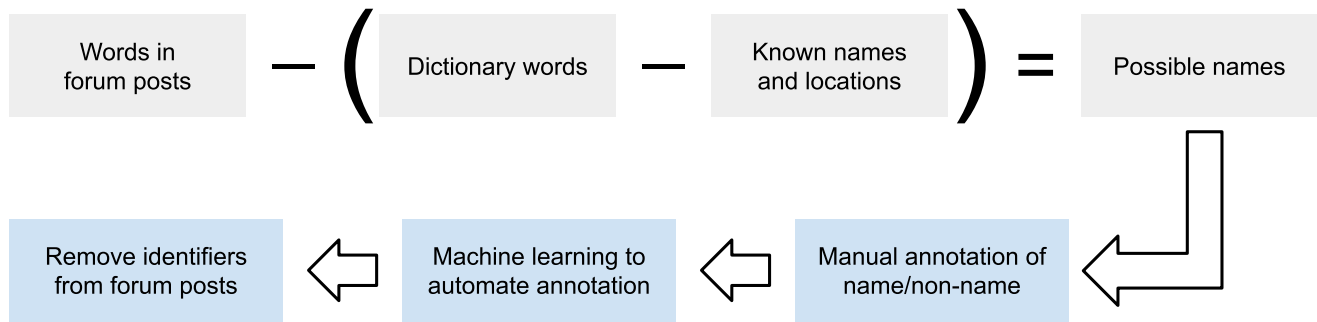


Figure 1. Anonymization method overview. Grey boxes indicate data, blue boxes indicate processing steps. Minus signs indicate set subtraction.

of data for course 2. In the first semester of course 1 there were 14,082 posts made by 226 individuals – including the instructor, whose identity and forum posts also need to be anonymized. In the second semester of course 1 there were 9,217 posts made by 295 individuals, and in course 2 there were 930 posts made by 78 individuals. Forum activities consisted of personal introductions, questions about and discussions of topics in the course, team formation/coordination for group projects, and others.

We developed the anonymization method by examining the largest dataset (the first semester of course 1, referred to as *training data*) and utilized the remaining two datasets as completely unseen held-out testing data. The second semester of course 1 (referred to as *holdout 1*) provides a test of generalization of the method over time, while course 2 (referred to as *holdout 2*) serves as a test with a different course topic, learning management system, and instructor.

We obtained approval from our institutional review board and the instructors of the courses before collecting and analyzing data. However, we were only permitted to access anonymized data for analyses. Hence, we developed the anonymization method in cooperation with university data warehouse staff, who ran code for analyses on original forum data and shared the anonymized results.

2.2 Narrowing the List of Possible Names

There are several possible categories for each word (sequence of consecutive non-whitespace characters, after removing punctuation) in a discussion forum post. The word may be an identifying name referring to a person or a place, an English word¹, a misspelling, or a non-English word (e.g., numbers, other languages). The most challenging and time-consuming aspect of anonymization is to determine whether a particular word is identifying or not. We applied a set-theoretic approach to drastically reduce the scope of the problem, narrowing down the list of all words in forum posts to a small subset of possible names, which are then much less time-consuming to annotate.

The top row of Figure 1 illustrates the possible name extraction process. We started with a dictionary of over 100,000 English words [36], including common loanwords, and removed any words that overlapped with a list of over 23,000 cities, political regions, and countries (words such as South, New, etc. that were part of place names)². We then also removed any words from the

dictionary that overlapped with a list of over 7,000 first and last names obtained from U.S. census data. Thus, the dictionary contained only words that were not the names of people or places – words like *wormhole* and *dalliance*, but not *so* or *will*. We then removed these non-name dictionary words from the list of all unique words in discussion forum posts, leaving only possible names.

2.3 Feature Extraction

We extracted various features to help both human annotators and machine learning models classify each possible name as a name or non-name word. Features can be categorized into two basic types: densely-distributed *ad hoc* features and sparsely-distributed word presence features. Ad hoc features calculated for each possible name consisted of:

- Count of occurrences
- Word index in the first post where the word was used
- Count of words in the first post where the word was used
- Proportion of occurrences where the word was capitalized
- Proportion of occurrences where the word was at the beginning of a sentence
- Proportion of mid-sentence occurrences (not at the beginning of a sentence) where the word was capitalized
- Proportion of occurrences where the word was mid-sentence and capitalized
- Whether the word was a dictionary word or not (before modifying the dictionary)
- Whether the word was in the U.S. census list of first/middle names
- Whether the word was in the U.S. census list of last (family) names
- Frequency of the word in the U.S. census list of first/middle names

¹ This paper focuses on English-language text. However, the method could be repeated for other languages by replacing English-specific components (e.g., the dictionary) with another language.

² Obtained from <http://www.geonames.org>

- Frequency of the word in the U.S. census list of last names
- Whether the word was in the list of world cities
- Whether the word was in the list of political regions (e.g., states, territories)
- Whether the word was in the list of countries
- Count of dictionary words that were within one edit (deletion, insertion, replacement, or transposition) of the word
- Count of dictionary words that were within two edits of the word

The list of ad hoc features resulted from several rounds of error analysis and iterative refinement, which was necessary to reach classification accuracies comparable to human raters. Feature development proceeded approximately in order of complexity. Our original features consisted of the simplest ideas such as the count of occurrences. Complex features, such as the proportion of capitalized mid-sentence occurrences, resulted from examining prediction errors from models with simpler features. While this process may have resulted in over-fitting the feature extraction process to training data, we made no adjustments to features for final evaluation on the holdout datasets (see description of dataset annotation below).

Word presence features indicated the presence or absence of a particular word within the 10 most common words preceding the possible name word, which we refer to as context words, among all of its occurrences across forum posts. We tracked the most common context words separately for capitalized mid-sentence occurrences, capitalized occurrences, and all occurrences. This separation helps to determine whether common dictionary words like “hope” were also names. Word presence features consisted of a 1 if a particular word appeared in the ten most common context words for the possible name in question, and a 0 otherwise. Word presence features captured things like if a word was preceded by “hi” or “hello” – words which tended to indicate the presence of a name. We limited these features to the 25 most common overall context words, yielding 75 total context word presence features (since there were 3 capitalization conditions). Additionally, we included an “other” count category for all less-common context words, yielding another three features (one for each capitalization condition). For example, the words “tea” and “coffee” might occur among the 10 context words for a particular possible name, but be too infrequent across all possible names to rank in the top 25; we would thus count these both as “other” and calculate features for the number of “other” words in the context words for that particular name, the number of capitalized “other” words, and the number of “other” words capitalized in the middle of a sentence. Thus there were 95 features in total: 17 densely-distributed and 78 sparsely-distributed.

2.4 Manual Annotation of Possible Names

Two raters iteratively annotated possible names derived from the training data, checked agreement, and updated an annotation scheme to resolve patterns of common disagreement. Annotators had access to features listed above, as well as the possible names themselves. They did not, however, have access to the actual forum posts nor to associated possible name pairs (first and last names together), thereby mitigating unnecessary exposure to possible identifying information.

In the first round, raters annotated 200 randomly-selected possible names as either names, non-names, or unknown. Of these 200, they annotated 10 as unknown. The annotation guide was subsequently revised to remove the unknown category, since ultimately a name/non-name decision must be made for anonymization, and to clarify unknown cases. Unknowns primarily consisted of famous individuals’ names (e.g., Obama), which we classified as names out of an abundance of caution. For the remaining 190 cases, the raters achieved 87.4% agreement and Cohen’s $\kappa = .734$ (confidence interval = [.634, .833]).

Raters annotated a different set of 200 randomly-selected possible names in the second round to test the updated annotation guide. They achieved 89.5% agreement and $\kappa = .773$ (confidence interval = [.681, .865]). After this round we added the mid-sentence capitalization features described above, to help disambiguate disagreements noted by the raters.

Raters completed a third round of annotation to test the final annotation guide, achieving 92.7% agreement and $\kappa = .842$ (confidence interval = [.820, .864]) on all 2,588 instances in the training data, indicating excellent agreement [7]. After this round they also annotated a sample of 650 randomly-selected possible names from the holdout 1 dataset, though we removed 50 of these when we later discovered that they were erroneously included due to UTF-8 encoding issues. This left a holdout sample of 600 possible names, which we deemed sufficient to produce a tight confidence interval for agreement, given the confidence intervals previously obtained with just 200 possible names. On the holdout 1 dataset the raters achieved 93.8% agreement, with $\kappa = .864$ (confidence interval = [.823, .907]), indicating that they were able to apply the annotation guide to a new dataset with at least as much agreement as the original dataset. Finally, raters discussed each of their disagreements to reach a definitive name/non-name label for each of the 600 possible name instances in holdout 1.

A single rater annotated a sample of 600 possible names in the holdout 2 dataset as well. Given the excellent agreement between raters, we deemed a single rater sufficient for this task. Specifically, the more conservative rater (higher recall; see rater comparison in Table 1 results) annotated the holdout 2 dataset.

The final data thus consisted of 2,588 labeled instances in the training dataset (35.5% annotated as names), 600 in holdout 1 (36.0% annotated as names), and 600 in holdout 2 (44.5% annotated as names), which we used to train and validate the automatic name classification procedure.

2.5 Name/Non-name Classification

The process of extracting possible names greatly reduces the burden of manual annotation and limits raters’ exposure to identifying information. We sought to further reduce these concerns by automating the classification step.

We evaluated two quite different machine learning approaches and ultimately combined them via decision-level fusion. The first classification algorithm was *Extra-Trees* [15], which is a variant of Random Forest that trains multiple trees (500 in our case) based on random subsets of data, and adds further randomness by choosing random points at which to divide the data in feature space. Extra-Trees makes no strict assumptions about data distribution, and thus works well for the features in this paper, which include densely- and sparsely-distributed features with vastly different ranges and distributions. Moreover, Extra-Trees has inherent feature selection (dimensionality reduction) capabilities, since irrelevant features can simply be ignored when constructing each tree. We utilized the

implementation of Extra-Trees available in *scikit-learn* for this model [30].

The second approach we evaluated was a deep neural network (DNN) implemented with *TensorFlow* using the stochastic gradient descent optimizer [1]. We developed a custom structure for the DNN to suit the specific properties of the problem (Figure 2). The feature space is relatively large (95 dimensions) for a model with no inherent dimensionality reduction capabilities, so we added regularization to constrain model complexity. The densely- and sparsely-distributed features call for different regularization methods, however. Several of the densely-distributed features were highly correlated, and the number of features (17) was relatively small. Thus, we applied L2 regularization for densely-distributed features [29]. Conversely, we applied L1 regularization to the sparsely-distributed features, of which there were many (78), since L1 pushes the weight of irrelevant features toward 0. We then concatenated the post-regularization outputs of fully-connected layers for densely- and sparsely-distributed features, and stacked additional fully-connected layers (which were regularized via dropout [31]). Finally, we added a fully-connected sigmoid activation output layer (i.e., logistic regression) to predict name or non-name.

We evaluated models via nested four-fold cross-validation on the training dataset. In this approach, we randomly selected 75% of instances (possible names), trained a model on those instances, and tested it on the remaining 25% of instances. We repeated the process three more times so that each instance was in the testing set exactly once. During training, we weighted false negative errors (incorrectly classifying a name as a non-name) twice as heavily as false positive errors (incorrectly classifying a non-name as a name), since we were more concerned about missing identifying information than about accidentally removing non-identifying words. False positive errors might adversely affect some analyses (e.g., if the word “joy” was mistaken for a name, thereby changing the result of sentiment analysis), but would not harm student privacy.

We tuned hyperparameters (model settings) for both models via nested cross-validation, in which we tested different hyperparameters and selected the best combination of hyperparameters based on cross-validated mean squared error. Note that this step took place nested within training data only, via 4-fold cross-validation within the training data of the outer 4-fold cross-validation loop, so that hyperparameters were not selected based on test set accuracy.

For the Extra-Trees model, we tested hyperparameters consisting of the minimum number of instances required for each leaf of the tree (values of 1, 2, 4, 8, 16, or 32) and the maximum proportion of features to consider when creating each tree branch (values of .25, .5, .75, or 1.0). For the DNN, we searched hyperparameters including the number of neurons in each hidden layer (2, 4, 8, 16, or 32), L2 regularization strength (.1, .01, or .001), L1 regularization strength (.1, .01, or .001), dropout regularization strength (0, .25, or .5), number of hidden layers after the concatenation layer (0, 1, 2, or 4), and the learning rate (.01, .001, or .0001). The hyperparameter search space consisted of the cross product of these values (i.e., grid search). Hence, training was time-consuming (several days), but the trained models can be applied to an entire course’s data in less than 10 seconds.

Finally, we re-trained the models on all training data and applied to the held-out dataset. We then combined model predictions to form a decision-level fusion model by simply averaging Extra-Trees and

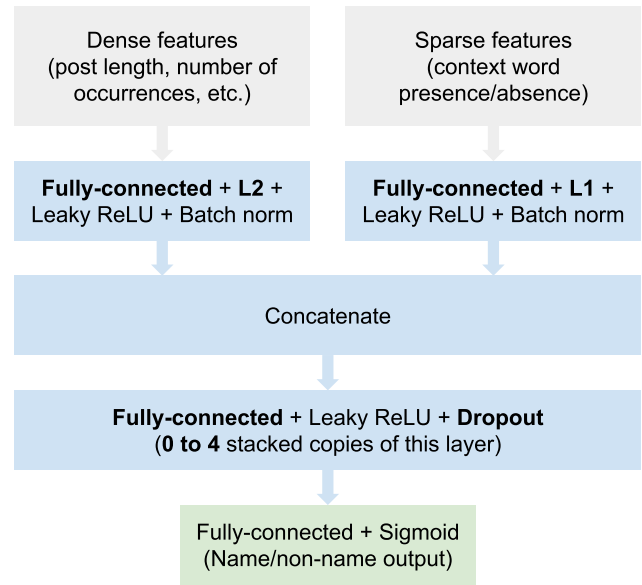


Figure 2. Custom DNN structure. Elements marked in bold were hyperparameters (number of neurons, regularization strength, or number of layers) tuned via nested cross-validation.

DNN predictions, for both cross-validated training set predictions and holdout predictions.

2.6 Removal of Identifiers

After names have been identified via manual annotation or automatic classification, removal of identifying information is relatively straightforward. First, we removed potential identifying information that follows known regular expression patterns, consisting of email addresses, URLs, phone numbers, and other numbers – which we removed in case they might represent things like social security numbers or other identifiers. As noted in previous research [35], such information can be identified with essentially 100% accuracy via pattern matching, and the challenging cases are names, nicknames, misspellings, and other name variants that we focus on in this paper. We replaced each pattern match with a placeholder (e.g., *phone_placeholder*) so that they could potentially serve as context words for name classification or be measured during analyses of forum content. Second, we replaced all identified name words with a placeholder, regardless of capitalization (see Error Analysis sections below for measures of how much this may result in non-name words being accidentally removed).

3. CLASSIFICATION RESULTS

We evaluated machine learning results in terms of common accuracy metrics below, but also compared human raters to evaluate the utility of the automatic approach as a replacement for manual annotation.

3.1 Machine Learning Accuracy

Table 2 contains key results of the automatic name classification method. Overall, results show that the models were highly accurate, reaching area under the receiver operating characteristic curve (AUC) as high as .981 in cross-validated evaluation on the training data, .979 on the holdout 1 dataset, and .956 on the holdout 2 dataset. AUC ranges from 0 to 1, where 1 indicates perfect classification

Table 2. Name vs. non-name classification results. FN indicates false negatives (names classified as non-names) and FP indicates false positives (non-names classified as names). Precision and recall refer to the positive (name) class. Acc refers to the percentage correctly classified.

Model	FN	FP	AUC	Acc	Cohen's κ	Precision	Recall	Base rate	N
<i>Cross-validated training data</i>									
Extra-Trees	52	202	.971	90.2%	.793	.811	.943	.355	2588
Custom NN	37	157	.981	92.5%	.841	.849	.960	.355	2588
Fusion	37	173	.980	91.9%	.828	.836	.960	.355	2588
<i>Holdout course 1 (later semester)</i>									
Extra-Trees	10	56	.976	89.0%	.772	.786	.954	.360	600
Custom DNN	11	55	.975	89.0%	.771	.788	.949	.360	600
Fusion	10	54	.979	89.3%	.778	.792	.954	.360	600
<i>Holdout course 2 (different course)</i>									
Extra-Trees	11	54	.950	89.2%	.784	.826	.959	.445	600
Custom DNN	16	61	.950	87.2%	.744	.805	.940	.445	600
Fusion	8	54	.956	89.7%	.794	.827	.970	.445	600

accuracy, 0 indicates completely incorrect classification, and .5 indicates random chance level. Thus, these results indicate models were accurate and generalized well from training data to the holdout data from 2 years later as well as the holdout data from another course.

We evaluated models with several different classification metrics in an effort to uncover any particular ways in which the method might be failing, but our primary concern was the number of false negatives – that is, the number of names misclassified as non-names. The decision-level fusion model yielded the lowest number of false negatives in each dataset (37, 10, and 8 for training, holdout 1, and holdout 2 respectively), and thus we intend to apply this model for practical use, though both of the individual models exhibited high accuracy as well.

The machine learning results also compare favorably to examples of previous work on anonymization of medical literature. In one study [28], researchers reported .967 recall on a training dataset (versus .960 for our decision-level fusion model) and .941 recall in a holdout testing set (versus recalls of .954 and .970 for our fusion model across the two holdout datasets). Moreover, our method did not require a dataset-specific list of names, as is common in previous work. While results are not exactly comparable, since base rates and predicted rates may have differed, they are strongly indicative of similar accuracy.

3.2 Comparison to Human Raters

Measuring annotation agreement between two human raters is one way to determine how “difficult” a task is, and whether a machine learning solution is close in accuracy. We computed machine learning model accuracy by comparing predictions to the resolved set of labels produced by raters; here we compare raters (pre-resolution) to each other. Since neither rater necessarily represents the ground truth more than the other, we computed comparison metrics alternately treating each rater as the ground truth.

Table 1 shows these results computed with the same accuracy metrics as the machine learning model, using the holdout 1 dataset.

Results show that the machine learning method was close to, and in some respects equally as accurate as the human raters. Recall and false negatives (FN) are especially important to consider for minimizing the risk of identifying information being revealed, and both showed that the fusion machine learning model (recall = .954, FN = 10) was close to or better than human accuracy depending on

Table 1. Details of human raters' agreement, treating each rater as ground truth individually to allow comparison to machine learning accuracy on the same task. AUC refers to the minimum proper AUC (calculated via linear interpolation with a single point) because raters provided only yes/no annotations, not probabilities.

	Ground truth	
	Rater 1	Rater 2
FN	30	7
FP	7	30
AUC	.945	.923
Acc (% agreed)	93.8%	93.8%
Cohen's κ	.864	.864
Precision	.864	.964
Recall	.964	.864
Base rate	.360	.360
N	600	600

which rater we considered as ground truth (recall .864 or .964, FN = 30 or 7). Note, though, that for cases where algorithmic recall exceeds human rater recall, human rater precision is correspondingly better since it is reciprocal with recall when treating a single rater as ground truth. In terms of κ , human raters do seem likely to be superior to the machine learning fusion model ($\kappa = .864$ versus .778). Thus, for some sensitive applications of the method, human raters may be needed. However, the fusion model primarily makes false positive (FP) errors, which are less of a privacy concern than FN errors.

The difference in FN between raters (FN = 7 versus 30) also indicates that there was some inconsistency in terms of tendency of one rater versus the other to make a classification of name or non-name. This is one potential advantage of making continuous-valued predictions (probabilities, in this machine learning case) of name versus non-name, because it allows setting a threshold. For human raters, thresholds are implicit but not easily or specifically controllable. As noted previously, we weighted false negative errors twice as heavily as false positive errors, though that is a parameter that could be adjusted for the particular needs of a dataset.

4. HOLDOUT 1 (LATER SEMESTER) ERROR ANALYSIS

We conducted an analysis of cases where machine learning predictions were incorrect, focusing on the decision-level fusion model applied to the holdout 1 dataset. Analysis of false negatives is important to discover the severity of cases where names are left unredacted, while analysis of false positives is important to quantify the amount of text that will be unnecessarily anonymized (replaced with placeholders).

4.1 False Negatives

False negatives are the most serious errors, since they may result in identifying information being revealed. There were 10 false negatives, which we examined to determine how serious these errors might be and to determine why they might have occurred. Human raters disagreed on 6 of the 10 words (and they only disagreed 37 times total – see Table 1), and only agreed to classify those 6 as names after discussing. This indicates that these were exceptionally difficult cases, even for humans. Furthermore, the machine learning method made similar false negative errors as human raters.

Of the 10 false negatives, there were 2 dictionary words (“long” and “mercy”), which may have indeed not been names. One of the 10 was the name of an entertainment company, which may have been an identifying characteristic (an employer) or, more likely, simply a reference to entertainment. Similarly, one was a name from a famous television show, and one was the name of a U.S. national park. The remaining words included a concatenated combination of words that was likely a filename but could have been a username, two non-English words, one name that seems likely to be a person’s first name (though it appeared only once in a forum post and was not capitalized), and one possible last name.

In sum, while there were several false negative predictions, examination of these cases reveals that even human raters initially disagreed for most of them and that it is quite possible that most of them, except probably the apparent last name, are indeed not names.

4.2 False Positives

While false positive errors are less serious, since they do not compromise identity, they do pose a challenge to subsequent analysis of forum text if important words are removed (e.g., words that might indicate sentiment, like “joy”).

The decision-level fusion model made 54 false positive errors. We observed several broad categories that capture most of these instances. First, we observed several geographical regions (e.g., “Africa”, “European”) that were too broad for our definition of identifying information – which was restricted to political regions – or even extraterrestrial (e.g., “Ganyemede”). Second, there were misspellings (e.g., “hellium” instead of “helium”), most of which were correctly identified as non-names but a few of which were not. Third, there were abbreviations such as “NBA” and “DOI”. Fourth, there were references to popular culture, such as “Overwatch” and “Kerbal”, which are indeed names but not identifying information. Finally, there were several domain-specific words, which we do not include as examples to avoid unintentional identification of the course from which data were collected.

Among these false positives, the most commonly-occurring word occurred just 26 times in 9,217 posts (the total size of the dataset from which holdout 1 data were sampled), most occurred only once, and all false positives combined appeared 191 times in those posts. This indicates that even though some non-name words were mistakenly removed from posts, the impact on the overall text was minimal.

5. HOLDOUT 2 (NEW COURSE) ERROR ANALYSIS

We performed similar analyses of classification errors for the holdout 2 dataset. However, it was not possible to compute inter-rater disagreement for the misclassified cases in holdout 2 because only one rater performed annotations.

5.1 False Negatives

There were just 8 FN errors among the 600 possible names in the holdout 2 dataset. Of these eight, three were abbreviations for university-specific terms, including a building name, a college (collection of university departments) name, and the name of a major. A further three FN errors were slang terms for large metropolitan areas with populations over 4 million. One was half of a misspelled two-word city name, and the last was a local street name.

None of the FN errors in this dataset were student names. The most serious errors are perhaps the university-specific terms, which could narrow down the identity of students when combined with other factors. However, in isolation (or even combined with each other) these terms match hundreds or thousands of students, and thus do not pose a likely risk for researchers hoping to analyze forum data.

5.2 False Positives

There were 54 FP errors in the holdout 2 dataset, which differ somewhat from the FP errors observed in holdout 1. Course 2, from which holdout 2 data were collected, utilized Roman numerals for assignment numbers, which were frequently mistaken for names. Additionally, the domain-specific content of course 2 required students to discuss a large number of letter combinations (strings) that do not represent words, and which were also often mistaken for names.

Like holdout 1, there were several misspellings mistaken for names in holdout 2 results. For example, “callender” (calendar), “hewlp” (help), and “ssolid” (solid) were FP errors. However, these account for very few redactions since these misspellings occurred only infrequently. The most notable FP was the word “my”, which occurred 293 times in the 930 posts in the holdout 2 dataset. Surprisingly, “My” was capitalized in 32.1% of its occurrences, including 15.4% of occurrences in the middle of sentences. This was somewhat unexpected, but appears to have frequently occurred when students did not punctuate the end of sentences and instead used line breaks (which we did not consider as end-of-sentence markers) to separate sentences.

Human intervention after the automatic classification step can easily correct false positive errors such as “my”, however. To facilitate this, our anonymization software produces a list of names identified by the machine learning fusion model as an intermediate step before the names are removed. The list includes the fusion model’s probability as well as the number of occurrences of each word in the original discussion forum data, sorted in descending order by occurrences. Researchers (or authorized staff in charge of anonymizing data) can thus easily examine the top of the list and delete any rows that are clearly high-impact FP errors before proceeding to the last step where names are redacted.

6. APPLICATIONS FOR INSTRUCTORS

Our primary motivation for developing this method was to enable research on the text of computer-mediated discussions students have with each other during their online learning experiences. However, such research may support the needs of course instructors as well, either directly via analysis methods they can easily apply, or via generalizable insights that can be applied to their courses. We thus sought out instructors of online university courses (who were not involved in the forum anonymization work or the courses analyzed in this paper) to gain better insight into instructor perspectives on scalable analysis of online course discussion forums. Specifically, we asked two instructors “*as an online course instructor, can you imagine any analyses of discussion forum text that would be informative for you?*”

6.1 Instructor 1

The first instructor was a male computer science faculty member with 14 years of university-level teaching experience, who had taught for-credit online courses at the university level as well as massive open online courses (MOOCs). He noted:

Specifically for all courses that I don't teach I don't have a legitimate need to know that student X is enrolled in course Y. Anonymization gives us a way to easily share forum discussions between different instructors of the same course, or across department etc. And there are numerous reasons why this is useful.

- * Potential for early detection of struggling students and the underlying cause. (Lack of time? interest? pre-reqs? effective strategies?)
- * Identification of hardest components of a course.
- * Research projects that look at common forum post across multiple courses. e.g. fresh/ sophomores/ seniors.

He also noted that when working with students to improve courses it is necessary to have anonymized data:

If I want to give the data to an undergrad staff for analysis for course improvement purposes (rather than for research publication), I'd require that they had anonymized data.

Additionally, instructor 1 conducts and publishes research on his own courses, and offered research questions and ideas he would like to pursue that would require anonymization. These included:

It may be possible to detect themes and generate hypotheses by skim reading the posts, but it is much harder to identify trends and quantitative trends (e.g. are there more X in the later part of the course). Also a general skimread of the forums will miss correlations with other data (e.g. students with background X tend to post more Y)

Can we identify when a course pace is too fast? Compared to assuming too much prior knowledge?

Suppose we consider a student's forum post action as an active intervention created by the student to affect on their own learning trajectory. How effective are these interventions? Do they also help similarly students that just read the discussion thread (and never need to post a similar issue themselves). Are they too late? Are they too early?

In sum, instructor 1 was enthusiastic about the prospect of being able to quickly anonymize online course discussion forums, and proposed several ways in which anonymization would benefit both teaching and research.

6.2 Instructor 2

The second instructor was a female statistics instructor and graduate student, with six years of university-level teaching experience. Her online courses are large, and thus provide unique challenges for teacher–student engagement. As she noted:

This semester there are about 1,400 students enrolled in [course information redacted]. It would be beneficial for me as an instructor to have some sort of automated analysis that told me which forums and topics were getting the most activity. That would help me know which forums to look at or have my undergrad course assistants look at and answer some of the questions. It would also be beneficial because if there was a lot of confusion about a certain topic, I would know I need to re-explain that topic in lecture.

I think something that identified negative words would be helpful too for the same reasons. If there's a lot of negativity on a thread- it's probably best that I go over that concept again in class to clarify any confusion.

While some of the needs noted by instructor 2 do not require access to the forum text itself (such as tools to measure forum activity), others would require researchers and developers to have access to anonymized forum text. For example, developing and validating methods for automatic assessment of confusion in forums is only possible with access to text data. Moreover, these needs highlight the difficulty of effectively utilizing online discussion forums with very large numbers of students, and the potential for automated tools to assist instructors in these courses.

7. DISCUSSION

In this study we were interested in enabling analysis of online discussion forums in university courses through removal of identifying information, even in cases where capitalization, grammar, and spelling may be unpredictable. Our results showed

that automatic anonymization is possible, and that it rivals human accuracy.

In this section we discuss implications of the results for various stakeholders, including users of the anonymization method (e.g., researchers, teachers) and students whose data is subject to analyses.

7.1 Implications for Users of the Anonymization Method

The proposed anonymization method offers two main advantages to users. First, it drastically reduces workload relative to approaches like manual identification and removal of identifying information directly from the forum text. Moreover, such manual anonymization is often intractable for users because they cannot access non-anonymized data in the first place. Second, it reduces users' exposure to identifying information. Users may either utilize the machine learning approach to avoid all involvement with identifying information, or annotate possible names manually – in which case they are still protected from seeing identifiers in the full context of the original text.

Anonymization is essential in many cases for researchers to either validate existing methods or develop new methods. For example, when automatically detecting sentiment from text with tools such as *SEANCE* (Sentiment Analysis and Social Cognition Engine; [9]) it is helpful to match sentiment to forum posts to obtain examples of the context in which sentimental language occurs. It is especially important to preserve student privacy in research that requires detailed reading of forum posts. For example, domain experts might annotate and evaluate the depth of questions students ask, or the responses they receive, to answer research questions about the relationship between a student's engagement with their peers and their status as a member of demographic groups that are traditionally-underrepresented in postsecondary education.

Finally, one important consideration for applications is how well the machine learning model is likely to generalize. We showed excellent generalization across time (2 years), as well as to a new course topic, instructor, and learning management system. While the change in topic (and instructor-specific course setup) did result in different types of errors, overall accuracy remained similar. However, we did not test across university populations. Students at other universities may have different backgrounds that influence how they interact with each other or with technology, and the vernacular language they use. Moreover, the same method could be applied to anonymize student-generated text in other contexts, such as college admissions essays [32], where students may reveal identifying information but in different (non-conversational) circumstances. Thus, for generalization to a notably different context, such as a different university or type of text, we recommend annotating a testing set of possible names to validate accuracy.

7.2 Implications for Students

The objective of our method is to minimize the potential for negative impacts on student privacy introduced by analyses of unstructured student-generated text. It is important, however, to recognize that such analyses carry inherent risk even with a (hypothetical) perfectly-accurate anonymization method. For example, students might mention their involvement in a particular

course in venues such as Twitter, Reddit, Facebook or others [39]. They may even post similar questions on course forums and public forums, or relate events that took place on course forums. It is unreasonable to expect perfect anonymization. Thus, it is important to take appropriate steps to limit public exposure to student data – even anonymized data – and to ensure that students reap benefits of analyses conducted on their data.

Positive impacts for students largely consist of 1) improvements made to future courses, and 2) additional capabilities afforded to instructors, both informed by research made possible through access to anonymized data. For example, researchers may be able to provide guidance to students about how to ask questions to elicit the most helpful responses. Or, as instructor 2 noted above, it might be possible to direct the attention of teaching assistants to students or topics where it is most needed.

Benefits to students are indirect in nature, and, in the case of research-informed changes to online courses, benefits might be more for future students than for the students from whom data were collected. Thus, more research is needed to sample student perspectives regarding analysis of their forum data, as well as their perspectives on the importance and impact of anonymization.

8. CONCLUSION

Access to discussion forum data is essential for researchers to better understand the experiences of students interacting with each other in web-based learning environments. However, access to these forum data is often hampered by important privacy concerns. Our approach for automatic anonymization of these data helps to resolve this issue, and has already enabled in-depth examination of forum posts [17–19, 38]. We plan to make our anonymization software publicly available³, and hope that it will be instrumental in advancing researchers' and teachers' knowledge of student experiences, and, ultimately improving learning in online classrooms.

9. ACKNOWLEDGMENTS

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A180211 to Board of Trustees of the University of Illinois. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

10. REFERENCES

- [1] Abadi, M. et al. 2016. TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* (2016), 265–283.
- [2] Anandan, B. and Clifton, C. 2011. Significance of term relationships on anonymization. *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 03* (USA, Aug. 2011), 253–256.
- [3] Anandan, B., Clifton, C., Jiang, W., Murugesan, M., Pastrana-Camacho, P. and Si, L. 2012. t-Plausibility: Generalizing words to desensitize text. *Transactions on Data Privacy*. 5, (2012), 505–535.

³ See <https://ilearn.illinois.edu> for anonymization software

- [4] Beckmann, J. and Weber, P. 2015. Cognitive presence in virtual collaborative learning: Assessing and improving critical thinking in online discussion forums. *Proceedings of the 2015 International Conference on E-Learning* (2015), 51–58.
- [5] Chakaravarthy, V.T., Gupta, H., Roy, P. and Mohania, M.K. 2008. Efficient techniques for document sanitization. *Proceedings of the 17th ACM conference on Information and knowledge management* (New York, NY, Oct. 2008), 843–852.
- [6] Chen, B., Chang, Y.-H., Ouyang, F. and Zhou, W. 2018. Fostering student engagement in online discussion through social learning analytics. *The Internet and Higher Education*. 37, (Apr. 2018), 21–30. DOI:https://doi.org/10.1016/j.iheduc.2017.12.002.
- [7] Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*. 20, (1960), 37–46. DOI:https://doi.org/10.1177/001316446002000104.
- [8] Crossley, S., Paquette, L., Dascalu, M., McNamara, D.S. and Baker, R.S. 2016. Combining click-stream data with NLP tools to better understand MOOC completion. *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge* (New York, NY, USA, 2016), 6–14.
- [9] Crossley, S.A., Kyle, K. and McNamara, D.S. 2017. Sentiment Analysis and Social Cognition Engine (SEANCE): An automatic tool for sentiment, social cognition, and social-order analysis. *Behavior Research Methods*. 49, 3 (Jun. 2017), 803–821. DOI:https://doi.org/10.3758/s13428-016-0743-z.
- [10] Cumby, C. and Ghani, R. 2011. A machine learning based system for semi-automatically redacting documents. *Proceedings of the Twenty-Third Innovative Applications of Artificial Intelligence Conference* (Aug. 2011).
- [11] Davis, G.M., Wang, C. and Yuan, C. 2019. N-gram graphs for topic extraction in educational forums. *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)* (2019), 532–535.
- [12] Deming, D.J., Goldin, C., Katz, L.F. and Yuchtman, N. 2015. Can online learning bend the higher education cost curve? *American Economic Review*. 105, 5 (May 2015), 496–501. DOI:https://doi.org/10.1257/aer.p20151024.
- [13] Dougiamas, M. and Taylor, P. 2003. Moodle: Using learning communities to create an open source course management system. (2003), 171–178.
- [14] Dwork, C. 2008. Differential privacy: A survey of results. *Theory and Applications of Models of Computation* (Berlin, Heidelberg, 2008), 1–19.
- [15] Geurts, P., Ernst, D. and Wehenkel, L. 2006. Extremely randomized trees. *Machine Learning*. 63, 1 (Apr. 2006), 3–42. DOI:https://doi.org/10.1007/s10994-006-6226-1.
- [16] Harrak, F., Bouchet, F., Luengo, V. and Bachelet, R. 2019. Automatic identification of questions in MOOC forums and association with self-regulated learning. *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)* (2019), 564–567.
- [17] Henricks, G.M., Perry, M. and Bhat, S. in press. Gender and gendered discourse in two online STEM courses. *Proceedings of the 14th International Conference on Learning Sciences (ICLS 2020)* (Nashville, TN, in press).
- [18] Huang, E., Valdiviejas, H. and Bosch, N. 2019. I’m sure! Automatic detection of metacognition in online course discussion forums. *Proceedings of the 8th International Conference on Affective Computing and Intelligent Interaction (ACII 2019)* (Piscataway, NJ, 2019), 241–247.
- [19] Jay, V., Henricks, G.M., Bosch, N., Perry, M., Bhat, S., Williams-Dobosz, D., Angrave, L. and Shaik, N. in press. Online discussion forum help-seeking behaviors of students underrepresented in STEM. *Proceedings of the 14th International Conference on Learning Sciences (ICLS 2020)* (Nashville, TN, in press).
- [20] Jiang, R., Banchs, R.E. and Li, H. 2016. Evaluating and combining named entity recognition systems. *Proceedings of the Sixth Named Entity Workshop, joining with 54th ACL* (2016), 21–27.
- [21] Kleinberg, B., Mozes, M., Arntz, A. and Verschuere, B. 2018. Using named entities for computer-automated verbal deception detection. *Journal of Forensic Sciences*. 63, 3 (2018), 714–723. DOI:https://doi.org/10.1111/1556-4029.13645.
- [22] Kortemeyer, G., Albertelli, G., Bauer, W., Berryman, F., Bowers, J., Hall, M., Kashy, E., Kashy, D., Keefe, H., Behrouz, M.-B., Punch, W.F., Sakharuk, A. and Speier, C. 2003. The learning online network with computer-assisted personalized approach (LON-CAPA). *Computer Based Learning in Science (CBLIS 2003)* (2003), 119–130.
- [23] Lee, S.Y., Chae, H.S. and Natriello, G. 2018. Identifying user engagement patterns in an online video discussion platform. *Proceedings of the 11th International Conference on Educational Data Mining (EDM 2018)* (2018), 363–368.
- [24] Meystre, S.M., Friedlin, F.J., South, B.R., Shen, S. and Samore, M.H. 2010. Automatic de-identification of textual documents in the electronic health record: A review of recent research. *BMC Medical Research Methodology*. 10, 1 (Aug. 2010), 70. DOI:https://doi.org/10.1186/1471-2288-10-70.
- [25] Nadeau, D. and Sekine, S. 2007. A survey of named entity recognition and classification. *Linguistic Investigations*. 30, 1 (Jan. 2007), 3–26. DOI:https://doi.org/10.1075/li.30.1.03nad.
- [26] Nanda, G. and Douglas, K.A. 2019. Machine learning based decision support system for categorizing MOOC discussion forum posts. *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)* (2019), 619–622.
- [27] Narayanan, A. and Shmatikov, V. 2009. De-anonymizing social networks. *2009 30th IEEE Symposium on Security and Privacy* (Piscataway, NJ, May 2009), 173–187.
- [28] Neamatullah, I., Douglass, M.M., Lehman, L.H., Reisner, A., Villarroel, M., Long, W.J., Szolovits, P., Moody, G.B., Mark, R.G. and Clifford, G.D. 2008. Automated de-identification of free-text medical records. *BMC Medical Informatics and Decision Making*. 8, 1 (Jul. 2008), 32. DOI:https://doi.org/10.1186/1472-6947-8-32.

- [29] Ng, A.Y. 2004. Feature selection, L1 vs. L2 regularization, and rotational invariance. *Proceedings of the Twenty-first International Conference on Machine Learning (ICML 2004)* (New York, NY, 2004), 78–85.
- [30] Pedregosa, F. et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*. 12, (Nov. 2011), 2825–2830.
- [31] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*. 15, 1 (2014), 1929–1958.
- [32] Stone, C., Quirk, A., Gardener, M., Hutt, S., Duckworth, A.L. and D’Mello, S.K. 2019. Language as thought: Using natural language processing to model noncognitive traits that predict college success. *Proceedings of the 9th International Conference on Learning Analytics & Knowledge* (Tempe, AZ, USA, Mar. 2019), 320–329.
- [33] Sweeney, L. 2002. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*. 10, 05 (Oct. 2002), 571–588. DOI:<https://doi.org/10.1142/S021848850200165X>.
- [34] Sweeney, L. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*. 10, 5 (2002), 557–570. DOI:<https://doi.org/10.1142/S0218488502001648>.
- [35] Sweeney, L. 1996. Replacing personally-identifying information in medical records, the Scrub system. *Proceedings of the AMIA Annual Fall Symposium* (Philadelphia, PA, 1996), 333–337.
- [36] Ubuntu – Details of package wamerican in bionic: 2017. <https://packages.ubuntu.com/bionic/wamerican>. Accessed: 2018-06-08.
- [37] Uijl, S., Filius, R. and Ten Cate, O. 2017. Student interaction in small private online courses. *Medical Science Educator*. 27, 2 (Jun. 2017), 237–242. DOI:<https://doi.org/10.1007/s40670-017-0380-x>.
- [38] Valdiviejas, H. and Bosch, N. in press. Using association rule mining to uncover rarely occurring relationships in two university online STEM courses: A comparative analysis. *Proceedings of the 13th International Conference on Educational Data Mining (EDM 2020)* (in press).
- [39] Wu, J.-Y., Hsiao, Y.-C. and Nian, M.-W. 2018. Using supervised machine learning on large-scale online forums to classify course-related Facebook messages in predicting learning achievement within the personal learning environment. *Interactive Learning Environments*. 28, 1 (Sep. 2018), 1–16. DOI:<https://doi.org/10.1080/10494820.2018.1515085>.

Predicting Engagement in Video Lectures

Sahan Bulathwela, María Pérez-Ortiz, Aldo Lipani, Emine Yilmaz and John Shawe-Taylor
University College London, United Kingdom
m.bulathwela@ucl.ac.uk

ABSTRACT

The explosion of Open Educational Resources (OERs) in the recent years creates the demand for scalable, automatic approaches to process and evaluate OERs, with the end goal of identifying and recommending the most suitable educational materials for learners. We focus on building models to find the characteristics and features involved in context-agnostic engagement (i.e. population-based), a seldom researched topic compared to other contextualised and personalised approaches that focus more on individual learner engagement. Learner engagement, is arguably a more reliable measure than popularity/number of views, is more abundant than user ratings and has also been shown to be a crucial component in achieving learning outcomes. In this work, we explore the idea of building a predictive model for population-based engagement in education. We introduce a novel, large dataset of video lectures for predicting context-agnostic engagement and propose both cross-modal and modality specific feature sets to achieve this task. We further test different strategies for quantifying learner engagement signals. We demonstrate the use of our approach in the case of data scarcity. Additionally, we perform a sensitivity analysis of the best performing model, which shows promising performance and can be easily integrated into an educational recommender system for OERs.

Categories and Subject Descriptors

K.3.1 [Computer Uses in Education]: Distance learning; H.3.1 [Content Analysis and Indexing]: Linguistic processing

General Terms

Human Factors, Measurement, Management

Keywords

Context-free Engagement, Cold Start, Video lectures, Quality Assurance, Open Education, OER, Personalisation

Sahan Bulathwela, María Pérez Ortiz, Aldo Lipani, Emine Yilmaz and John Shawe-Taylor "Predicting Engagement in Video Lectures" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 50 - 60

1. INTRODUCTION

With the recent popularity of online learning platforms, the creation of Open Educational Resources (OERs) is increasing rapidly [16]. This recent large-scale creation of educational material demands for ways to automatically manage educational resources. In the context of OERs, this means finding and recommending material that fits the learners' goals while maximising learning outcomes. Such a goal usually entails a large personalisation factor. We define it as *contextualised engagement*, which captures how engaging a learning resource is with regard to the context of the learner (e.g., learning needs/goals and learner state). Although contextualised engagement has gained interest in the recent years [8], we argue that there is also a *context-agnostic engagement* factor, that only relates to features of the learning resource and attempts to capture the gold-standard label of population-based engagement (i.e. the marginal of contextual engagement for a resource across the population of learners). Modelling context-agnostic engagement enables identifying highly engaging resources across a population of learners before personalising educational recommendations to individuals. This paper studies the features involved in context-agnostic engagement, as a first step towards building an integrative educative recommendation system, that will join both contextualised and context-agnostic features [9].

A high quality learning resource needs to satisfy three main properties: i) academic soundness and appropriate coverage of the body of knowledge, ii) pedagogical robustness and iii) enabling learners to achieve their desired learning outcomes [24]. Learner engagement has been shown to be a proxy for (iii), as engaging with material is a prerequisite for learning. There is evidence from both online [33, 23] and classroom [30, 36] educational settings showing that higher learner engagement increases the likelihood of better learning outcomes. We thus focus on finding the general characteristics of engaging material. Using features that can be extracted across multiple modalities (video, text, audio etc.) allows developing prediction models for gold-standard engagement that are easily adaptable to a wide range of OERs and can be automated [27].

Our work is one of the first to address educational engagement prediction with video lectures, specially from a quantitative perspective. One of our primary goals is to understand if easily automatable cross-modal features can be used as predictors for how engaging an educational resource is, as

opposed to modality specific features. Although large-scale studies (involving millions of videos) have been conducted to analyse the prediction of engagement for general purpose videos [40], the largest study in the context of educational video lectures involves 800 videos from 4 courses and analyses engagement from a qualitative perspective [17]. To the best of our knowledge, this work is the first attempt to predict engagement with educational videos automatically. Our experiments involve more than 4000 video lectures that span over 20 diverse subjects, making it the largest dataset to date in this field. Our dataset, code and best performing model are released with the paper.

Given the usefulness of *predicting context-agnostic engagement* and the scarcity of work in this topic, we are motivated to answer the following research questions, which will enable the deployment of such a model in an educational platform:

- RQ1 How to encode context-agnostic engagement?
- RQ2 How effective are cross-modal language-based features for predicting engagement with video lectures?
- RQ3 Does including modality-specific features lead to a significant improvement in performance?
- RQ4 What features influence context-agnostic engagement?
- RQ5 Is predicting marginal population-based engagement useful over personalised engagement?
- RQ6 Can we assume a common underlying model for predicting engagement across different knowledge areas?

2. RELATED WORK

The interest in identifying useful and engaging information goes beyond the educational domain and is investigated in numerous other fields [10]. For example, Wikipedia uses a review system to evaluate the quality of its articles. To do so, different machine learning models, such as support vector regression and ensemble methods, are used with features such as text style, readability, structure, network, recency and review information [14, 39]. Moreover, in the context of automatic essay scoring, promising results have been obtained through rank preference support vector machines [41] and more sophisticated deep learning models [37].

Quality-based document ranking [3] and spam web-page detection [28] are other areas in the information retrieval domain that also utilises textual features and recency related features. These features categorise into different verticals such as understandability, topic coverage, presentation, freshness and authority [10].

OERs available to the public come in large-scale and various modalities [27, 19], which makes modality-specific models of limited use. As existing work proposes models with domain/modality specific features (e.g. network features of Wikipedia [15] or speaker speed in videos [17]), there is a need for models that can evaluate how engaging educational materials are at scale using a cross-modal feature set. We attempt to address this gap through this work.

2.1 Why Modelling Engagement?

As argued by Lane [24], a well designed learning resource should enable the learner to achieve the expected learning outcomes. Prior work has studied *learner engagement* in Massively Open Online Courses and shown that when optimised, **engagement can increase the likelihood of achieving better learning outcomes** [33, 23]. User engagement has also been shown to differ greatly from popularity measures such as number of views [40], as the latter does not necessarily capture whether learners consume the material. In our work, we also show that engagement does not positively correlate with user ratings. Instead, what we observe is that lectures with low rating also present low engagement rate. However, lectures with greater ratings can have different engagement rates.

For videos, *watch time* has been used as the main measure for quantifying engagement in the literature, e.g., for YouTube recommendations [13], predicting engagement with videos [40]. For educational content, the median of normalised engagement time (i.e., the percentage of watch time from the total video) has been used as gold standard for engagement [17]. Our work tests several approaches to encoding user engagement.

Most of the related work regarding predicting educational engagement attempts to model learner engagement as a function of the learner’s context (demography, user activity, etc.) [4, 19, 2], as opposed to modelling context-agnostic learner engagement as a function of content-based features of the educational resource, which is our aim. Context-agnostic engagement has been previously studied for video lectures, advocating for qualitative and general recommendations such as keeping videos short [17], using conversational language for lecture delivery [5] and others. These recommendations empower authors to create better educational videos. However, none of these works address the need for automatically identifying the features of highly engaging educational resources, which is imperative for retrieving and recommending educational material at scale.

3. DATA AND METHODOLOGY

This section first describes the dataset built for predicting engagement, together with the set of features proposed in this paper. Then, we introduce the machine learning methods and the feature importance analysis method considered.

To address the research questions outlined in the introductory section of this paper we do the following: i) We study different ways of refining user engagement signals, linking to literature on psychometrics (RQ1). ii) We propose two sets of easily automatable features for predicting engagement (cross-modal features inspired by context-agnostic quality literature and video-specific features) and evaluate the difference of predictive performance between them (RQ2 and RQ3). iii) We construct a large dataset of video lectures and evaluate the performance of the proposed engagement signals and sets of features (RQ2-4). iv) We compare cross-modal to modality specific features, analysing the impact of individual features in the predictive model that presents the most promising performance (RQ4). v) We compare our population-based engagement approach to its personalised analogue to demonstrate its usefulness (RQ5). vi) We

compare the engagement models obtained from dividing the video lectures in two differentiated knowledge areas: STEM (such as technology, physics and mathematics lectures) vs others (such as arts, social science and philosophy lectures).

3.1 Dataset and Features (RQ2-4)

We use data from a popular OER repository, VideoLectures.Net (VLN)¹, a collection of videos of researchers presenting in peer-reviewed conferences. This data is suitable for our aim for two reasons: i) It contains watch patterns about how learners consume lectures, and ii) the lectures are peer-reviewed and hence material is controlled for correctness of knowledge and pedagogical robustness. The transcriptions of English lectures and English translations for the non-English lectures are provided by the TransLectures project². We restrict the final dataset to lectures that has been viewed by at least 5 unique users, leading to the final dataset having 4,063 lectures. These lectures are categorised into 21 subjects, e.g. Computer Science, Physics, Philosophy, etc. Learner engagement labels of the dataset is computed using 155,850 user view log events (video viewing events) created between December 8, 2016 and February 17, 2018. The dataset constructed is publicly available, including different statistics of population engagement and all the cross-modal and video-based features proposed.

3.1.1 Cross-modal Features

We selected a subset of cross-modal and mostly language-based features that are easy to extract from the VLN dataset. The 13 extracted features are shown in Table 1. This set has been selected based on recurring features in the related work [3, 14, 17, 28, 39] and their quality verticals [10] identified in our prior work. The majority of features were extracted using methods and token (word) sets that are found in the prior work referenced in Table 1.

Additionally, we introduce the *published date*, represented by converting the video publication date to UNIX epoch time (in days). In other words, it is the number of days between January 01, 1970 and the lecture published date.

3.1.2 Video-based Features

We also extracted four out of the seven features proposed for analysing educational engagement with video lectures from [17], selecting those features that can be automatised and are objective. These are: i) *lecture duration*, as shorter videos have been shown to be much more engaging; ii) *is chunked*, whether the lecture has been partitioned into multiple parts; iii) a set of indicator variables describing the *type of lecture*, such as tutorial, workshop, etc; and iv) *speaker speed*, measured by the average amount of words spoken per minute. We also include the *silence period rate (SPR)*, calculated using the special tags in the video transcripts that indicate silence. Formally, for a lecture ℓ , this feature $\text{SPR}(\ell)$ is calculated as follows:

$$\text{SPR}(\ell) = \frac{1}{D(\ell)} \sum_{t \in T(\ell)} D(t) \cdot \mathcal{I}(N(t) = \text{"silence"}), \quad (1)$$

where t is a tag in the collection of tags $T(\ell)$ that belong to lecture ℓ , N returns the type of tag t and D returns the du-

¹www.videolectures.net

²www.translectures.eu

Table 1: Extracted features from the VLN dataset.

Feature	Reference
<i>Content-based features</i>	
Easiness (FK Easiness)	[14]
Stop-word Presence Rate	[28]
Stop-word Coverage Rate	[28]
Document Entropy	[3]
Word Count	[39]
Title Word Count	[3]
Preposition Rate	[14]
Auxiliary Rate	[14]
To Be Rate	[14]
Conjunction Rate	[14]
Normalization Rate	[14]
Pronoun Rate	[14]
Published Date	—
<i>Video-based features</i>	
Lecture Duration	[17]
Is Chunked	[17]
Video Lecture Type	[17]
Speaker speed	[17]
Silence Period Rate (SPR)	—

ration of tag t or lecture ℓ and $\mathcal{I}(\cdot)$ is the indicator function (returning 1 when the condition is verified, 0 otherwise).

3.2 Quantifying Engagement (RQ1)

Our work focuses on implicit user feedback (most specifically, engagement). Implicit feedback (in the form of number of views, engagement or any other measure that does not require the user to provide explicit feedback) has been used for building recommender systems for nearly two decades with great success [29, 20, 22], as an alternative to explicit ratings, which have a high cognitive load on users and thus are usually sparse. However, implicit signals have other challenges associated with them. For example, implicit feedback is usually positive-only [20] and can contain effects such as popularity bias, i.e., there might be a bias towards more popular items, whereas implicit feedback for other items may be very sparse. There has been several works investigating the relationship between explicit and implicit feedback [12, 34, 42], which we also do through this work.

The main measure that we use to quantify engagement is the **Median of Normalised Engagement/watch Time (MNET)**, as it has been proposed as the gold standard for engagement with educational materials in previous work [17]. To have the MNET label in the range [0, 1], we set the upper bound of MNET to 1. We observed in our initial data analysis that MNET values in the VLN dataset follow a Log-Normal distribution, where it can be seen that most users generally abandon the lecture after a generally low time threshold. We hypothesise this may be because it takes some time to decide whether the content is relevant for the learner. Users that make it after this threshold seem more committed and thus the leaving rate is significantly lower. To address this, as this is usually a problem when using machine learning methods, we applied a log transformation to transform the engagement signal. The final label, *Log Median Normalised Engagement Time (LMNET)*

is computed using the following:

$$\text{LMNET}(\ell) = \ln(\max(\text{MNET}(\ell), 1)). \quad (2)$$

To test if LMNET can be further improved, we compare this approach of encoding engagement to other alternative ways of quantifying and cleaning engagement signals, drawing inspiration from the literature on psychometrics and subjective assessment [21, 38], which focuses on explicit human feedback and assumes that users present cognitive biases and differences, with applications in preference ranking and measuring perception-based qualities, such as engagement. The intuition behind this is that different learners may have a different engagement threshold and scale, similarly as with explicit ratings [21]. We compare different approaches for defining engagement:

1. **Raw LMNET**, as per Eq. (2) which considers that no user differences exist and the marginal over the population can be directly used as gold standard label for engagement, similarly as in [17].
2. **Cleaned LMNET**, for which we test the removal of bot-like users (those users with an average engagement rate less than 5%), which may have a detrimental factor in the median of raw engagement.
3. **Standardised LMNET**, in which we preprocess LMNET per user (subtracting the mean of the user and dividing by the standard deviation), as commonly done with human ratings in order to remove user biases and differences [21]. In this scale, positive values indicate lectures that are more engaging than the mean of the user and vice versa.
4. **Comparative MNET**, in which we exploit the law of comparative judgement and use psychometric scaling to go from user comparative engagement data to a probabilistically interpretable engagement scale [38, 32]. More specifically, we assume that engagement data can only be compared per user (as users may have different biases, thresholds or engagement scales). To do so, we generated a matrix of engagement comparisons (of the type: Did learner i prefer lecture A to B in terms of engagement?), which is used as the input for psychometric scaling, producing a final scale in which distances can be interpreted in terms of probability of greater engagement.

As discussed, the limitation of these approaches is that they disregard the context of the learner and the temporal component that may inherently be present when engaging with educational material. A different measure to encode engagement is found in Wu et al. [40], where the main idea is to compare engagement relative to the length of the video. The authors propose this for entertainment videos. However, we argue against this approach in the case of educational material, as the aim is to take the learner to the desired state in the most efficient way, thus the general recommendations found in the literature of keeping videos as short as possible [17].

3.3 Machine Learning Models (RQ2)

To learn to rank video lectures based on engagement, we evaluate the performance using pointwise ranking models. Regression algorithms predict the target variable in real value space ($y \in \mathbb{R}$), which allows them to create a global ranking of observations based on predictions. We also evaluate the performance of engagement prediction using kernelised models. Kernelisation allows capturing non-linear patterns in data without having to operate in the respective basis. Although it is more computationally efficient than working in the non-linear space itself, it is more computationally expensive than solving the non-kernelised problem. Our choice of kernel for the models is the Radial Basis Function (RBF). RBF kernel is widely used in the literature and has mathematical connections to other popular kernels such as exponential and polynomial kernels [11, 35].

We use two regression algorithms, namely, *Ridge Regression (RR)* and *Support Vector Regression (SVR)* in primal form. We use RR as it is a widely used algorithm for regression [40] and SVR as it has performed well in a similar task in prior work [14]. We also evaluate the performance of the kernelised version of the same two algorithms (with RBF kernel), *Kernelised Ridge Regression (KRR)* and *Kernelised Support Vector Regression (KSVR)*. This allows us to understand if there is non-linearity in the patterns that benefits the prediction task. In all four models discussed above, we employ standard scaling as these models are not scale invariant. L2 regularisation is used to defend against overfitting and multicollinearity [26]. As ensemble techniques have shown to perform well in prior work [39], we also employ a *Random Forest Regressor (RF)* to evaluate its prediction capabilities. This model is also capable of capturing non-linear patterns.

3.3.1 Comparison to Personalised Models (RQ5)

One of our aims is to compare the population-based model to its personalised counterpart. The idea in this case is to test if a common baseline can be assumed for all users. For this, we train the same machine learning models per user, using the features previously proposed.

3.4 Feature Importance Analysis (RQ4)

Understanding how different features influence engageability of materials is vital in educational domain as learners will be guided on life-changing pathways based on these judgements. In a conventional linear model such as RR or SVM, feature importance analysis is straightforward as the weight coefficients reflect the influence of features.

In this paper we use *SHapley Additive exPlanations (SHAP)*, which is a model-agnostic framework that quantifies the impact of features on the model predictions. It reliably estimates feature importance of complex model families such as ensembles [25]. A SHAP value is computed for every feature of every prediction. Given a prediction and a feature, SHAP is computed by averaging how the prediction changes when the feature is present and vice versa. This procedure enables quantifying the contribution of each feature to the model prediction. By plotting all the SHAP values of the prediction data points in a SHAP summary plot, we can identify how each feature influences the prediction. By calculating the *Mean Absolute SHAP (MAS)* for each feature

f over the observations:

$$\text{MAS}_f = \frac{1}{N} \sum_{n=1}^N |\text{SHAP}_{f,n}|, \quad (3)$$

we obtain a more quantitative understanding of feature influence. N is the number of observations.

4. EXPERIMENTS AND DISCUSSION

This section shows the experimental setup and results for the different experiments conducted.

4.1 Experimental Setup

The evaluation of the machine learning models is performed using a 5-fold cross-validation for both feature sets. The performance of different machine learning models with different engagement quantification approaches can be found in Table 2. The performance when video-specific features are added is found in Table 3.

After gaining an understanding of model performance (see results in Table 2), we employ the best performing method and encoding for the rest of the analyses, using a hold-out validation with a train-test split of 70:30 to save computation. That is, the model is trained on the 70% training set and interpreted using the 30% test set. The experiments were implemented using `Scikit-learn` [31], `textatistic` [18] and `SHAP` [25] python packages. The source code in python and dataset are publicly available³.

4.1.1 Evaluation metrics

Pairwise accuracy (Pair.) and *Spearman Rank Order Correlation Coefficient (SROCC)* are the ranking metrics we used to evaluate the ranking performance of machine learning models with different engagement signal encodings.

Identifying models that can rank between video lectures is the core objective of this work. Hence, we devise *pairwise accuracy* as the main evaluation metric. Pairwise accuracy is more intuitive for this task as it represents the fraction of pairwise comparisons where the model could predict the more engaging lecture. Another opportunity that pairwise comparison provides is the ability to restrict the comparisons to subsets of lecture pairs (e.g. lectures that belong to the same subject, lectures that have similar LMNET).

In some of our experiments we also perform misranking analysis and report the pairwise accuracy. Misranking could happen if a subset of examples is systematically difficult to rank. We hypothesize that misclassification happens more frequently as the difference of LMNET between a pair of video lectures gets smaller. That is, the model may struggle to differentiate between two lectures with similar engagement. By doing this analysis, we can also understand the sensitivity of the prediction model to similarly engaging lectures. Obviously, misranking a pair of lectures that are significantly different in engagement incurs a larger cost in terms of user satisfaction than misranking a pair of lectures with similar engagement.

³<https://github.com/sahanbull/context-agnostic-engagement>

4.1.2 Controlling for Topics in Content

The topics covered in the content of the lecture is likely to drive learner engagement. For instance, Data Science lectures can be more popular than Physics lectures leading to easy pairwise comparison predictions between the domains. To test this, we restrict in some experiments the pairwise accuracy calculation to pairs of lectures that belong to the same domain (*subject-specific* column in Table 3) and observe if the accuracy value changes significantly compared to its counterpart metric that considers all lecture pairs in a domain-agnostic fashion.

4.2 Results

This section presents a series of experiments to:

- E1 Analyse the relationship between engagement, number of views and mean star ratings (RQ1).
- E2 Test different machine learning models and engagement signals for the cross-modal features (RQ1-2).
- E3 Study the distribution of engagement with respect to length of materials (RQ4).
- E4 Study the influence of modality-specific features and comparison across subject areas (RQ3).
- E5 Analyse the importance of different features in the model (RQ4).
- E6 Compare the population-based model to its personalised counterpart (RQ5).
- E7 Test if the same underlying model can be assumed for different knowledge/subject areas (RQ6).

4.2.1 E1: Engagement vs Views and Ratings

The VLN data source also has mean star ratings (explicit feedback) for a subset of the considered lectures. It is noteworthy that we only have access to mean star ratings, not to the individual ratings per observer or the number of measurements. As done in previous work, we also analyse the relationship between implicit signals (engagement and number of views) and explicit ratings. This can be found in Figure 1, where we show mean star rating vs MNET and number of views. The SROCC is close to zero, mainly because of the large number of lectures with high rating but low engagement and number of views. We test the correlation for the 4 different versions of engagement considered (raw, cleaned, standardised and comparative), but all achieve similar results, with SROCC close to zero. One conclusion that is clear from the plot in Figure 1 is that number of views, ratings and engagement do represent very different information. For example, it can be appreciated that the variance of MNET and number of views increases with higher ratings, showing heteroskedasticity. This indicates that for low quality resources (with low ratings) engagement is generally low, whereas for resources with higher ratings engagement differs and may be either high or low. This suggests other factors involved in engagement than simply quality perceived by learners. Regarding number of views it seems that the correlation is rather negative, showing that the materials with the highest number of views present very low engagement.

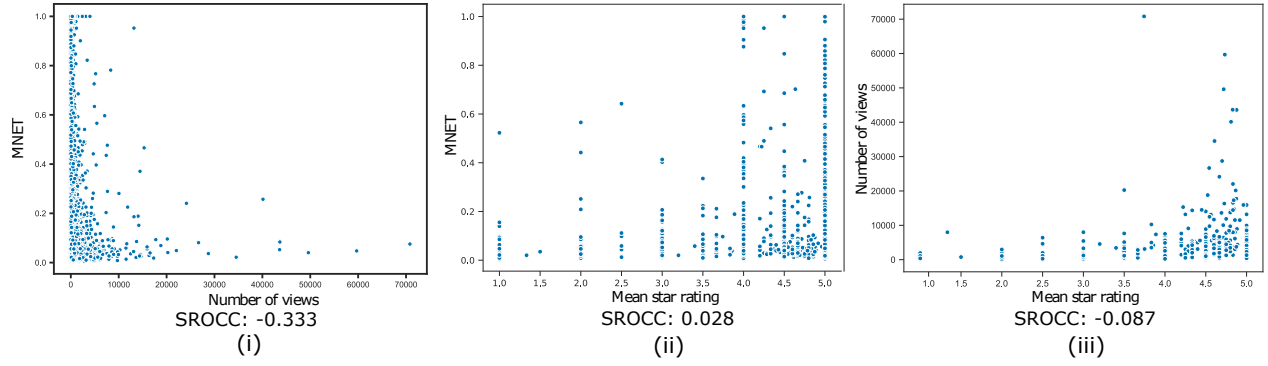


Figure 1: Scatter plots showing the relationship between (i) number of views vs. MNET, (ii) mean star rating for the video lecture vs. MNET and (iii) mean star rating vs. number of views, together with the Spearman’s rank correlation coefficient (SROCC).

Table 2: Pairwise accuracy (Pair.) and Spearman’s Rank Correlation Coefficient(SROCC) of engagement prediction models with standard error from 5-fold cross validation and cross-modal features.

Model	RR		SVR		KRR		KSVR		RF	
Engagement	Pair.	SROCC	Pair.	SROCC	Pair.	SROCC	Pair.	SROCC	Pair.	SROCC
Raw	.705±.011	.581±.027	.707±.000	.586±.000	.715±.004	.607±.011	.714±.007	.604±.019	.723±.009	.625±.027
Clearned	.636±.033	.396±.093	.634±.031	.392±.089	.646±.025	.424±.071	.642±.028	.414±.078	.646±.031	.427±.087
Standard	.603±.035	.302±.098	.600±.035	.292±.100	.609±.035	.315±.099	.602±.025	.297±.071	.611±.035	.323±.099
Comparative	.624±.010	.365±.028	.624±.012	.363±.036	.626±.013	.370±.040	.627±.009	.373±.027	.636±.012	.397±.038

4.2.2 E2: Encoding and Predicting Engagement

Inherently, the task of finding a better engagement signal is very challenging, given the lack of ground truth. In this paper, we first attempt to see if any of these signals present better correlation with star ratings. However, we observe from Figure 1 that engagement is not strongly correlated with perceived quality by users (explicit star ratings) and similar results emerge for different methods of quantifying engagement, meaning it is inconclusive that transforming raw engagement signals strengthens its relationship to explicit perceived quality. Thus, in order to decide on which is the best way of capturing and quantifying engagement, we compare the pairwise accuracy for the four proposed approaches (raw LMNET, cleaned, standardised and comparative). This simply tells us which output target variable is easier to predict given the proposed features. Table 2 presents these results, together with the pairwise accuracy (Pair.) and Spearman’s Rank Order Correlation Coefficient (SROCC) obtained for each machine learning model with the standard error bounds based on 5-fold cross validation. The larger the accuracy value, the better performing the model is.

These results suggest that raw LMNET may be the most appropriate target label, particularly since the proposed features seem to be more useful when building a model for predicting raw LMNET. These results do not contradict the literature, both educational and non-educational, as MNET has been used as the gold-standard way of quantifying engagement. Our experiments thus showed that the use of subjective assessment inspired transformations do not improve the predictive power of engagement signals. This may be because these transformations/correction methods are initially designed to address biases in latent user preferences.

Although similar biases may exist in learners when consuming educational materials (e.g. learner fatigue, different engagement thresholds, language level preferences, etc.) we hypothesise that the most influential driver of engagement is the information content and style of the video.

Another observation from Table 2 is that KRR and KSVR models outperform their linear versions. This suggests that there could be non-linearity in the dataset that is better captured by the kernel techniques. RF seems to be the best performing model providing more evidence that non-linearity plays a significant role.

To show how the accuracy changes when the difference of MNET between two lectures changes, we first compute all the possible differences between pairs of lectures and binarize these pairs into bins of size 0.1 from 0 to 1, finally we compute the pairwise accuracy for each bin. Figure 2 shows how the performance of the model changes based on the *difference of MNET between lecture pairs*. The bars in the figure represent the pairwise accuracy for all the pairs that belong to the same bin. For example, the pairs with largest difference of MNET are predicted correctly with 0.962 accuracy whereas pairs with the smallest difference are predicted with 0.642 accuracy.

Intuitively, a learner might have a similar experience consuming a pair of video lectures that are similarly engaging (at least disregarding the topic), as one is less likely to notice the difference. The black line in Figure 2 presents the *cumulative pairwise accuracy* of the model if we were to assume that the learners are insensitive to noticing the difference of experience for lecture pairs that have a small difference of MNET. The plotted cumulative pairwise accuracy (y-axis)

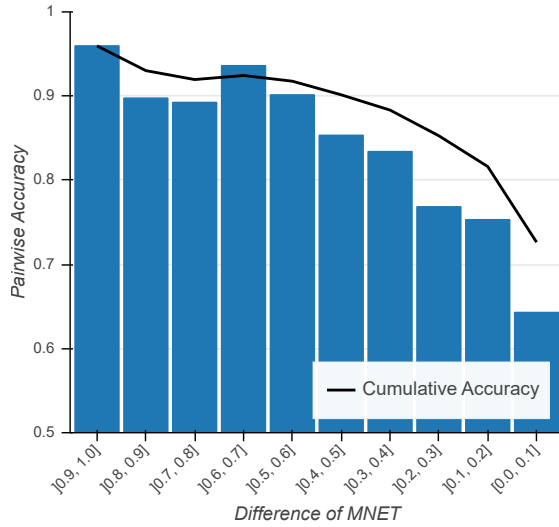


Figure 2: Bar chart plot showing how the pairwise accuracy changes based on the difference of MNET between lecture pairs

is computed by restricting the comparisons to lecture pairs with a difference of MNET between the lower bound of the x-axis value and 1.0. For instance, the cumulative pairwise accuracy of the model is 0.816 when the learners do not notice the difference when interacting with similarly engaging lecture pairs with MNET difference of $[0.0, 0.2]$. This value is the pairwise accuracy of all the lecture pairs with a MNET difference of $[0.2, 1.0]$.

4.2.3 E3: Length of Materials vs. Engagement

Several studies have shown that features that quantify material length have a significant impact (this is also reaffirmed by our observations in our feature importance analysis in Figure 6 and 7) on sustained engagement with the material [17, 14]. We investigate how the length of the lectures impacts engagement prediction (i.e. if the engagement predictor is naïvely distinguishing between long vs. short video lectures). We first investigate the distribution of total word count in the video lectures (Figure 3), which is directly related to the length. Based on the observed multi-modal distribution, we make two groups, i) short lectures of less than 5000 words and ii) long lecture (see engagement distribution in Figure 4). It can be seen that, as anticipated, the percentage of watch time tends to be shorter for long lectures.

We investigate how median engagement labels are distributed in the aforementioned groups and also how the pairwise accuracy differs among and between the groups. Figure 5 shows that the model is better at comparing between short-short lecture pairs compared to long-long lecture pairs. In the context of VLN dataset, this is good because there are more short lectures than long lectures (Figure 3). Recent findings (e.g.[17]) also encourage authors to make short videos, increasing the likelihood of future video productions being short lectures. MNET distribution in Figure 4 shows that long lectures have a more skewed target value distribution concentrated closer to 0 compared to short lectures

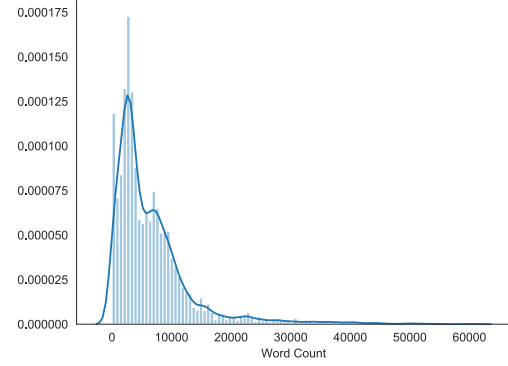


Figure 3: Distribution of word count of video lectures

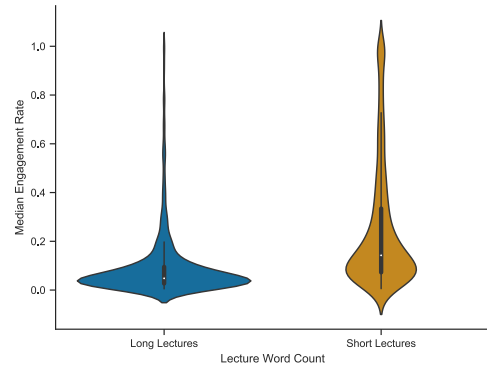


Figure 4: Distribution of engagement labels for short and long lectures.

Table 3: Pairwise accuracy with standard error via 5-fold cross validation for RF model using content-based features vs. content-based + video-specific features.

Model	Pairwise Accuracy	
	Subject-agnostic	Subject-specific
Content-based Features	.724±.014	.733±.018
Video-specific Features	.744±.011	.755±.014

suggesting that learners tend to consume smaller fractions of long videos. This is likely to be driven by factors beyond other measured features of the lectures, such as limited time availability and short attention span of learners.

4.2.4 E4: Video-Features and Subject Areas

Table 3 shows how the pairwise accuracy increases when restricted to subject-specific comparisons (lecture pairs belonging to the same subject area). This is clearly an advantage, given that most often, an educational recommendation system needs to make choices among sets of resources that belong to the same subject area.

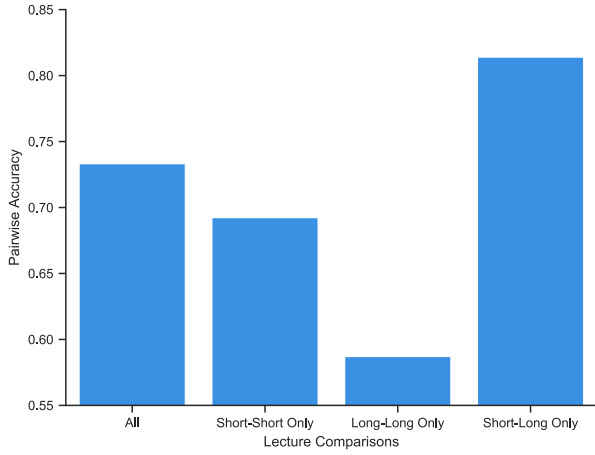


Figure 5: Accuracy bar chart for different types of comparisons using short and long lecture labels.

Table 4: Influence of content-based features on engagement as per their verticals outlined in [10].

Quality Vertical	Feature	MAS	% MAS
Topic Coverage	Word Count	.250	.366
Freshness	Published Date	.107	.157
Understandability	Easiness	.052	.076
Understandability	Stop-word Coverage Rate	.042	.061
Presentation	Normalization Rate	.039	.058
Topic Coverage	Title Word Count	.039	.057
Presentation	To Be Rate	.038	.055
Topic Coverage	Document Entropy	.033	.048
Understandability	Stop-word Presence Rate	.028	.041
Presentation	Conjunction Rate	.019	.028
Presentation	Preposition Rate	.014	.020
Presentation	Pronoun Rate	.013	.020
Presentation	Auxiliary Rate	.009	.013

Table 3 additionally shows how the performance differs when using exclusively the cross-modal set of features and when adding video specific features. The addition of video features increase the performance by approximately 2%. This result shows that there is a compromise in performance when restricting features to cross-modal features although the feature extractors can be reused in a practical scenario.

4.2.5 E5: Feature Importance Analysis

The SHAP value summary plots for content-based and video-specific feature sets are presented in Figures 6 and 7 respectively, where the features are ordered based on overall feature influence using the best performing prediction model (RF). Colour represents the raw feature value (blue low, red high). For example, when the observed values of a feature is red and they have a negative SHAP value, this means that higher values of this feature negatively impact LMNET prediction. Regarding video length, figures validate its impact on engagement, showing that long videos generally present lower engagement and vice versa, with lecture duration and word count being the most relevant features. Prior studies confirm this observation [17, 40, 15]).

Table 4 complements Figure 6 by giving a more quantitative representation of how the influence of different features

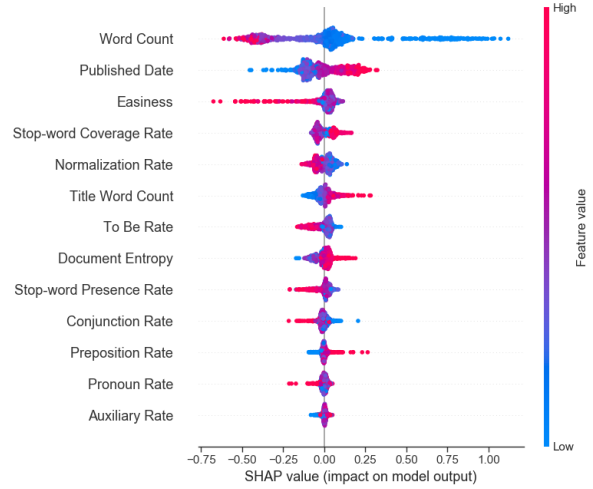


Figure 6: SHAP summary plot for cross-modal features.

across the test dataset changes. Higher MAS is associated with more important features. By looking at the five most influential features, we observe that all identified quality verticals (topic coverage, understandability, freshness and presentation) are represented. This observation supports the importance of considering all the different verticals when predicting context-agnostic engagement. The influence of top features is also consistent with results on quality biased information search [3] where it is also found that *Title Word Count* is comparatively less important. Figures 6 and 7 also show the importance of modality-specific features in this prediction task by raising *Lecture Duration*, *Silence Period Rate* and *Speaker Speed* in Figure 7 to high ranks.

4.2.6 E6: Population-based vs. Personalised

We use the 20 most active learners from the VLN dataset to compare the predictive performance of context-agnostic to contextual/personalised models when predicting engagement. Firstly, we train the population-based prediction model using the VLN dataset (outlined in section 3.1) using a 70:30 train-test split. In order to build the personalised model, for each user, we make a similar 70:30 train-test split respecting the temporal order of their individual events. We use the training data to build a personalised model per user using only the cross-modal set of features (no video-specific features). For each learner ℓ , we make predictions on the N_ℓ test events using (i) population-based model and (ii) the personalised model trained on personal events of the learner. We calculate Mean Absolute Error ($MAE(\ell)$) as:

$$MAE(\ell) = \frac{1}{N_\ell} \sum_{n=1}^{N_\ell} |y_n - \hat{y}_n|, \quad (4)$$

where \hat{y}_n is the prediction. As regression models are devised for the task, MAE is a sensible evaluation metric to measure predictive performance of the models. Then we calculate the difference of $MAE(\ell)$ between the population-based and personalised model. Thus, a negative value indicates that the population model is better and vice versa.



Figure 7: SHAP summary plot with video-specific features.

Figure 8, where the y-axis represents the difference in performance between the population-based and personalised, shows that the population-based model has better predictive power when the number of training examples available for the individual learner is limited (≈ 60). This is represented by the green line (at a MAE difference of 0). This demonstrates the usefulness of the population-based engagement prediction model in a situation where the recommender system is in a cold-start phase.

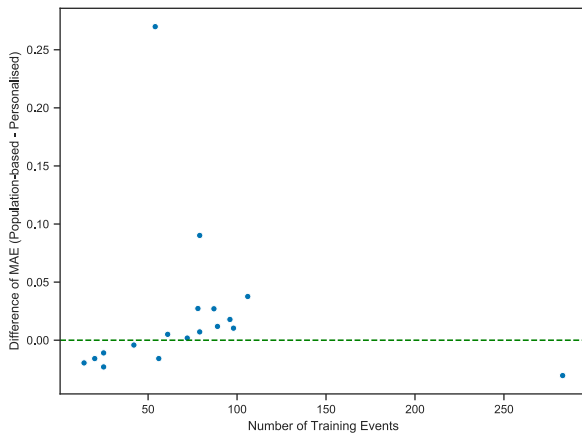


Figure 8: How the difference between Mean Absolute Error (MAE) of population-based and personalised models change with the number of training events per learner. Each data point is an individual learner in the dataset.

Table 5: Pairwise accuracy for STEM and Miscellaneous (Misc.) lectures when trained with subject-agnostic and subject-specific training data

Training Data	Test Data	
	STEM	Misc.
Subject-agnostic	.737	.708
Subject-specific	.732	.704

4.2.7 E7: Individual Models per Knowledge Area

To understand if training subject-specific models can improve on the predictive power of the overall task, we partition the lecture records into 2 categories:

- **STEM:** *Life Sciences, Physics, Technology and Mathematics.*
- **Miscellaneous:** *Social Sciences, Humanities, Arts and Philosophy.*

Then, we compare the performance of the models trained on subject-agnostic (STEM + miscellaneous) and subject-specific (STEM only or miscellaneous only) training data. Table 5 demonstrates that there is little evidence in our results contradicting that a common subject-agnostic engagement model can be assumed across knowledge areas. This is shown in the fact that both training with all knowledge areas or dividing into two, the models obtain very similar test accuracy for each category (.737 vs .732 and .708 vs .704). In fact, the best performance is obtained in both cases by training with the whole dataset. This indicates that in general a common engagement model can be assumed throughout knowledge areas.

4.3 Limitations

Firstly, the model does not include features that capture authority of content or its authors. Authority has been identified as an influential feature and lacking it is a weakness of this model. However, identifying an authority indicator that generalises beyond niche communities (e.g. academia) is challenging yet necessary, especially in the OER landscape where anyone can author learning materials. Additionally, the topic coverage features used in this model (*Word Count*, *Title Word Count* and *Document Entropy*) are relatively naïve, although they are useful. Having better features will likely improve the model. The current work demonstrates promise in predicting learner engagement with video lectures using easily automatable material features alone. More sophisticated features, both cross-modal and modality-specific could lead to higher predictive performance and better understanding of context-agnostic engagement. Thirdly, the engagement model is trained on English lectures and English translation of non-English lectures. This impacts the generalisation ability of the model. The same applies to non-video content as well. More rigorous testing is needed in these fronts. Lastly, given that our dataset only considers OERs and excludes the learning dimension, we highlight that some of our findings may not be directly applicable to other type of educational material. Particularly, given that most of our features are language-based and we disregard visual information, the built models may not generalise to general purpose videos.

5. CONCLUSIONS

Given its timely need, we set out to develop and empirically test the suitability of engagement prediction models for automatically assessing context-agnostic engagement of OERs. Due to the scarcity of publicly available datasets for the task, we sourced a new video-lectures dataset and evaluated how different machine learning models perform on this dataset. In our analysis, we observed that the Random Forest algorithm performs best. We show that cross-modal features provide satisfactory performance, which is a major advantage, since these can be extracted from different resource modalities. Further experiments show that the predictive performance of the model can gain a slight boost in performance by adding modality-specific features. However, the performance does not deviate significantly. Feature analysis showed that lecture length features are the most influential features in predicting context-agnostic engagement, which agrees with prior work. Other moderately influential features come from diverse quality verticals. Our analysis also showed that the model classifies much better when lectures with very different engagement values are compared, as opposed to lectures with similar engagement. This is natural and obviously the negative impact of misranking pairs of similar engagement lectures is relatively small. Our experiments demonstrated that the built model is useful in data scarcity scenarios, e.g. to approach the common cold-start problem in recommender systems. This is both for new users and new content, as our model can automatically estimate the engagement for new material and the model can be used as a prior for when we do not have enough data from a user to build a personalised model. We finally show that dividing the dataset into different knowledge areas (Subjects) and building separate models does not show improved performance, thus validating that a common underlying model can be built for estimating engagement across differentiated knowledge areas.

The proposed context-agnostic engagement prediction model can be beneficial in improving different components of an educational recommendation system. In situations where new content is discovered frequently (e.g. OER landscape [27, 7]), the proposed prediction model estimates *how engaging materials* are prior to exposing them to the learner population. This allows better balancing the risks relating to learner satisfaction with opportunities of having fresh materials. Also, the proposed context-agnostic model can be integrated with a personalisation system in different ways. It can act as a prior that mitigates cold-start problem both on user and content fronts. In systems where personalisation heavily focuses on the topics covered in the materials [9], this model can complement the content-based model by accounting for stylistic and lingual features that go beyond topic coverage.

To further improve the models, future work should address the three main limitations discussed: Future versions of our model should incorporate more sophisticated features. It could be beneficial to include features capturing *authority* and *topic coverage* [10]. In this sense, Wikification [6] can be used to extract covered topics, and data driven authority features, such as [1], can be used to learn a universal author authority score. In the cross-modal front, more features focusing on content understanding, such as topic coherence

and argument strength, can be considered. In the video-specific front, features such as liveliness of the presenter, sound quality and narration quality can be incorporated. Regarding the generalisation capabilities of the model, evaluating the effectiveness of the cross-modal feature set with a bigger video lecture dataset [17, 40] and a text dataset [14] will increase the confidence on the feature set. Similarly, non-English datasets should also be taken into account.

6. ACKNOWLEDGMENTS

This research is part of the X5GON project funded from the EU's Horizon 2020 research programme grant No 761758 and partially funded by the EPSRC Fellowship titled "Task Based Information Retrieval", under grant No EP/P024289/1.

7. REFERENCES

- [1] B. T. Adler and L. de Alfaro. A content-driven reputation system for the wikipedia. In *Proc. of Int. Conf. on World Wide Web*, 2007.
- [2] C. R. Beal, L. Qu, and H. Lee. Classifying learner engagement through integration of multiple data sources. In *Proc. of AAAI Conference on Artificial Intelligence*, 2006.
- [3] M. Bendersky, W. B. Croft, and Y. Diao. Quality-biased ranking of web documents. In *Proc. of ACM Int. Conf. on Web Search and Data Mining*, 2011.
- [4] F. Bonafini, C. Chae, E. Park, and K. Jablolkow. How much does student engagement with videos and forums in a mooc affect their achievement? *Online Learning Journal*, 21(4), 2017.
- [5] C. J. Brame. Effective educational videos: Principles and guidelines for maximizing student learning from video content. *CBE—Life Sciences Education*, 15(4), 2016.
- [6] J. Brank, G. Leban, and M. Grobelnik. Annotating documents with relevant wikipedia concepts. In *Proc. of Slovenian KDD Conf. on Data Mining and Data Warehouses (SiKDD)*, 2017.
- [7] S. Bulathwela, S. Kreitmayer, and M. Pérez-Ortiz. What's in it for me? augmenting recommended learning resources with navigable annotations. In *Proceedings of the 25th International Conference on Intelligent User Interfaces Companion*, IUI '20, 2020.
- [8] S. Bulathwela, M. Pérez-Ortiz, R. Mehrotra, D. Orlic, C. de la Higuera, J. Shawe-Taylor, and E. Yilmaz. Sum'20: State-based user modelling. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 899–900, New York, NY, USA, 2020. Association for Computing Machinery.
- [9] S. Bulathwela, M. Perez-Ortiz, E. Yilmaz, and J. Shawe-Taylor. Towards an integrative educational recommender for lifelong learners. In *AAAI Conference on Artificial Intelligence*, AAAI '20, 2020.
- [10] S. Bulathwela, E. Yilmaz, and J. Shawe-Taylor. Towards Automatic, Scalable Quality Assurance in Open Education. In *Workshop on AI and the United Nations SDGs at Int. Joint Conf. on Artificial Intelligence*, 2019.
- [11] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin. Training and testing

- low-degree polynomial data mappings via linear svm. *Journal of Machine Learning Research*, 11(Apr):1471–1490, 2010.
- [12] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *Proceedings of the 6th International Conference on Intelligent User Interfaces*, IUI '01, pages 33–40, New York, NY, USA, 2001. ACM.
- [13] P. Covington, J. Adams, and E. Sargin. Deep neural networks for youtube recommendations. In *Proc. of ACM Conf. on Recommender Systems*, 2016.
- [14] D. H. Dalip, M. A. Gonçalves, M. Cristo, and P. Calado. Automatic assessment of document quality in web collaborative digital libraries. *Journal of Data and Information Quality*, 2(3), Dec. 2011.
- [15] D. H. Dalip, M. A. Gonçalves, M. Cristo, and P. Calado. A general multiview framework for assessing the quality of collaboratively created content on web 2.0. *Journal of the Association for Information Science and Technology*, 2017.
- [16] M. Ehlers, R. Schuwer, and B. Janssen. Oer in tvet: Open educational resources for skills development, 2018.
- [17] P. J. Guo, J. Kim, and R. Rubin. How video production affects student engagement: An empirical study of mooc videos. In *Proc. of the First ACM Conf. on Learning @ Scale*, 2014.
- [18] E. Hengel. Publishing while Female. Are women held to higher standards? Evidence from peer review. Cambridge Working Papers in Economics 1753, 2017.
- [19] M. Hussain, W. Zhu, W. Zhang, and S. M. R. Abidi. Student engagement predictions in an e-learning system and their impact on student course assessment scores. *Computational Intelligence and Neuroscience*, 2018(6347186), 2018.
- [20] D. Jannach, L. Lerche, and M. Zanker. Recommending based on implicit feedback. In *Social Information Access*, 2018.
- [21] L. Janowski and M. Pinson. The accuracy of subjects in a quality experiment: A theoretical subject model. *IEEE Transactions on Multimedia*, 17(12):2210–2224, Dec 2015.
- [22] G. Jawaheer, P. Weller, and P. Kostkova. 8 modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Transactions on Information and System Security*, 2:26 pages, 05 2014.
- [23] A. S. Lan, C. G. Brinton, T.-Y. Yang, and M. Chiang. Behavior-based latent variable model for learner engagement. In *Proc. of Int. Conf. on Educational Data Mining*, 2017.
- [24] A. Lane. Open information, open content, open source. In *The Tower and The Cloud*, pages 158–168. 2010.
- [25] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*. 2017.
- [26] A. Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proc. of Int. Conf. on Machine Learning*, 2004.
- [27] E. Novak, J. Urbančič, and M. Jenko. Preparing multi-modal data for natural language processing. In *Proc. of Slovenian KDD Conf. on Data Mining and Data Warehouses (SiKDD)*, 2018.
- [28] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proc. of Int. Conf. on World Wide Web*, 2006.
- [29] D. W. Oard and J. Kim. Implicit feedback for recommender systems. In *Proceedings of the 1998 AAAI Workshop on Recommender Systems*, 1998.
- [30] Z. A. Pardos, R. S. Baker, M. O. San Pedro, S. M. Gowda, and S. M. Gowda. Affective states and state tests: Investigating how affect and engagement during the school year predict end-of-year learning outcomes. *Journal of Learning Analytics*, 1(1), 2014.
- [31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2011.
- [32] M. Perez-Ortiz and R. K. Mantiuk. A practical guide and software for analysing pairwise comparison experiments, 2017.
- [33] A. Ramesh, D. Goldwasser, B. Huang, H. Daume III, and L. Getoor. Learning latent engagement patterns of students in online courses. In *Proc. of AAAI Conference on Artificial Intelligence*, 2014.
- [34] B. Shapira, M. Taieb-Maimon, and A. Moskowit. Study of effectiveness of implicit indicators and their optimal combination for accurate inference of users interests. *JDIM*, 4(3):169–174, 2006.
- [35] J. Shawe-Taylor, N. Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.
- [36] S. Slater, R. Baker, J. Ocumpaugh, P. Inventado, P. Scupelli, and N. Heffernan. Semantic features of math problems: Relationships to student learning and engagement. In *Proc. of Int. Conf. on Educational Data Mining*, 2016.
- [37] K. Taghipour and H. T. Ng. A neural approach to automated essay scoring. In *Proc. of Conf. on Empirical Methods in Natural Language Processing*, 2016.
- [38] L. L. Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273–286, 1927.
- [39] M. Warncke-Wang, D. Cosley, and J. Riedl. Tell me more: An actionable quality model for wikipedia. In *Proc. of Int. Symposium on Open Collaboration, WikiSym '13*, 2013.
- [40] S. Wu, M. Rizoio, and L. Xie. Beyond views: Measuring and predicting engagement in online videos. In *Proc. of the Twelfth Int. Conf. on Web and Social Media*, 2018.
- [41] H. Yannakoudakis, T. Briscoe, and B. Medlock. A new dataset and method for automatically grading esol texts. In *Proc. of HLT*, 2011.
- [42] P. Zigoris and Y. Zhang. Bayesian adaptive user profiling with explicit & implicit feedback. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management, CIKM '06*, pages 397–404, New York, NY, USA, 2006. ACM.

The Ebb and Flow of Student Engagement

Measuring motivation through temporal pattern of self-regulation

Steven C. Dang
Carnegie Mellon University
stevenda@cs.cmu.edu

Kenneth R. Koedinger
Carnegie Mellon University
koedinger@cmu.edu

ABSTRACT

Effective teachers recognize the importance of transitioning students into learning activities for the day and accounting for the natural drift of student attention while creating lesson plans. In this work, we analyze temporal patterns of gaming behaviors during work on an intelligent tutoring system with a broader goal of detecting temporal trends in students' motivation. Findings demonstrate that observing gaming the system behaviors in the near beginning or end of a working session correspond with predictions made by self-regulation theories of ego-depletion and task-switching. Furthermore, analyses provide initial evidence these gaming behaviors are indicative of partial cognitive engagement and session-level influences on student motivation. These findings provide evidence for how temporal fluctuations in students motivations might be inferred through self-regulated behaviors like gaming the system, and how such information could inform better more intelligent tutoring systems that are responsive to cognitive and motivational dynamics during student work.

Keywords

Motivation, Self-Regulation, Measurement, Gaming the system, Ego-depletion, Task-switching, Intelligent tutoring system

1. INTRODUCTION

Many teachers can relate to the struggle of keeping an entire class engaged as the end of the day approaches. Some students may be listening raptly while other have started packing their belongings. Many teachers use class management techniques, such as specific activities in the beginning of class, in anticipation of the difficulties in ramping up the engagement of the entire class [9]. Student motivation appears to vary systematically over the course of a class period. Many good teachers adapt to this reality. It seems appropriate that intelligent tutoring systems should as well.

Student procrastination, the failure to engage in a task in a timely fashion, has a well-established link to student motivations [16]. The nature of the tasks that students have difficulty engaging themselves in can be revealing about their individual goals [15], their perceptions of the value of the task [5], and their beliefs about their abilities to complete the task [21]. Similarly, the context of what drives students to quit can be equally telling about

the same facets of student motivation [20].

Measures of quitting and procrastination leverage the easily observable dichotomy of student engagement, but are there other within-task student behaviors that might similarly indicate motivation? Quitting and procrastination are evidence of students' failure to exercise their self-regulation. In these moments, students are failing to direct their attention towards a less desirable but beneficial learning task, and instead opting to engage in more desirable non-learning tasks. Applying this self-regulation lens, it may be possible to understand student motivation by identifying and analyzing other observable moments during student work where students engage in less desirable behaviors for learning.

1.1 Temporal Dynamics of Self-Regulation

Self-regulation is the capacity to control or direct one's attention, thoughts, emotions, and actions [27]. One of the leading models of self-regulation poses the construct as a reward-based decision-making process [2]. In this model, self-regulation is treated as a series of decisions that seeks to optimize some expected value based on anticipated rewards and costs. Motivation is defined as "the orienting and invigorating impact, on both behavior and cognition, of prospective reward" [2]. Through this theoretical lens, self-regulation decisions are a reflection of student's motivation.

For instance, solving an extra credit problem on the homework may likely push the student's grade from a B to an A for the year. However, the problem will likely take an hour to solve and the student may have to skip soccer practice to find time to complete the problem. Observing the student's choices and behaviors in these critical moments of self-regulation can reveal student's underlying motivation. Prior models of self-regulated learning behavior have focused on the cognitive facets of a given task: its difficulty level [4,7], its domain topic[10], its time cost[6], and its expected value to the student[19]. However, research on self-regulation point to temporal factors that influence decision making.

Task switching research indicates that the exercise of self-regulation imposes a cognitive cost. Once an individual chooses to engage in a task, they do not always appear to be applying themselves with full effort [8]. Additionally, when a person is forced to change tasks rapidly, they are not able to perform at the same level as those given more consolidated spans of time to perform on the same task [11]. These studies imply that students are likely to perform at a reduced capacity when initially beginning work to perform on a task upon initially beginning work,

Ego-depletion models of self-regulation posit that the ability to regulate attention over time may tend to deplete as some time-driven function of an internal and limited resource [23]. Thus,

Steven Dang and Kenneth Koedinger "The Ebb and Flow of Student Engagement: Measuring motivation through temporal pattern of self-regulation" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 61 - 68

motivation may also tend to wane over time leading to an eventual failure to self-regulate.

In this work, we seek to investigate whether these temporal properties of self-regulation are evident in the prevalence of student's failures to self-regulate.

2. Related Works

Measuring self-regulation related constructs is not a new concept in the intelligent tutoring system literature. Prior work has developed a range of models for detecting self-regulation related behaviors.

2.1 Off-task Detection

Some of the earliest work in this space identified off-task student behaviors by identifying large gaps of time between interactions in the log data of student interactions [26]. Inferences on student skill improvement, in addition to whether the students asked for help or attempted a problem correctly/incorrectly following a long gap between interactions determined whether students were off-task while idle.

[18] developed models of mind-wandering, when students' attention and thoughts move off-task, which enabled detection of off-task behavior over much shorter time spans. These models leveraged information from videos and human labels of short time segments to train a supervised model to classify when mind wandering occurs. The features fed into the model included a range of low-level image processing features, facial features, inferred emotions, and temporal features that describe the dynamics of facial features and emotions during a short time interval. [17] extended this work given user self-reports of mind-wandering and included body position information.

2.2 Persistence and Quitting

[4] developed a model of student persistence by analyzing patterns of behavior that included observed student actions contingent on properties of the problems being worked and the student's skill on those problems. In this work, two types of students emerged, where the authors posited that trait level differences in students' capacity for sustained attention lead to differences in learning strategies and persistence during problem solving.

[3] designed a game-based measure of trait level persistence and validated the measure against other existing survey and standard psychometric behavioral tasks. The measure looked at average time on unsolved versus solved problems given a wide range of difficulty levels.

In [7], the authors built models of quitting an educational game. They leverage many features including features of each level of the game, the current state of game progress of the student, and the time in the current level. The final model that emerged from the supervised machine learning process were focused around actions of the student and the state of progress and counts of actions at each level across and within attempts at the level, thus not including any of the limited temporal features given at model training time.

[10] attempted to predict when students would quit reading a given passage. In this work, the authors used semantic features of the reading passages, the recent context of what passage is being read, which passages have been read recently, and both current page and total reading time. Total reading time, a similar proxy to ego-depletion, was found to be a significant contributor to models of quitting with respect to the first page of a passage. The authors

also implicitly investigated the role of task switching by predicting quitting at the beginning of a new passage compared to some other new page within a passage. While some of the data supports a differential impact of task switching and time on quitting, the authors do not explicitly explore how quitting behaviors vary over time.

2.3 Gaming the System

With intelligent tutoring systems that provide scaffolding supports through progressively informative hints and feedback, another behavior tends to arise called "gaming the system" [24]. These behaviors have been identified using information about a series of recent actions such as time spent or the number of recent hint requests and errors, and the characteristics of the problems worked, such as problem section and difficulty in those interactions [12]. Extensive work has attempted to determine what drives gaming behaviors. While some initial work determined that problem context better explained gaming behaviors over trait-like individual propensities to game [25], later work presented the opposite result using a different intelligent tutoring system [22]. A large multi-environment analysis was conducted that compared the types of gaming behaviors observed across urban, suburban, and rural contexts using three different intelligent tutoring systems [13]. The study found that across tutoring environments, students displayed different predominant gaming behaviors, which implies that the lure of certain types of gaming may be different given tutoring environment or problem-type affordances. Similarly, within tutoring environments, students from areas of different population density (eg: rural versus urban) display different predominant patterns of gaming. These differences point to how variation in work environment may have differential anticipated costs to gaming, while the variation within environment but across geographic regions point to possible cultural and thus motivational differences.

2.4 Research Questions

Prior work has developed extensive models of self-regulation behaviors that demonstrate the importance of cognitive, contextual factors, and local temporal factors for influencing student's self-regulation decisions. However, these models have not investigated how self-regulation behaviors might vary systematically over time and how such trends relate to student learning. In this work, we seek to investigate whether the within-session temporal properties of self-regulation are evident in student behaviors and whether these temporal trends are predictive of similar negative impacts on student learning.

Models of the cognitive cost of task switching imply that self-regulation related behaviors such as gaming the system are more likely to occur in the beginning of a work session. Similarly models ego-depletion imply that self-regulation related behaviors such as gaming are more likely to occur after students have been working for some time. We propose to investigate whether models of task-switching and ego-depletion are evident in some changes over time of the probability of gaming the system, a behavioral instance of self-regulation. We then investigate whether lower cognitive engagement as predicted by task-switching theory co-occurs with gaming the system. We follow this with an analysis to determine if failures in self-regulation during critical time periods are indicative of session-level motivation.

3. The Dataset

We utilize an observational dataset [1] including 214 students across 22 classrooms using the Carnegie Learning Cognitive Tutor (CT) in Pre-Algebra, Algebra 1, and Geometry. The tutor

was used approximately two class-periods per week for a full school year. The dataset includes over 2.3M user transactions covering 55, 33, and 26 curricular units divided into 173, 98, and 44 sections across the three courses respectively.

The CT leverages computational cognitive models to provide adaptive problem selection and hint support and correctness feedback to the students. Problems are broken down into a multi-step process, which allows the system to identify independent skills and trace skill improvement over a fine-grained skill model of the domain. On each step, the system is able to provide multiple levels of hint support, with the final level containing the answer to the problem step. The system logs all interactions with the system including problem attempts, hint requests, response accuracy, and problem step time. In this study, transactions for all students over the course of an entire academic year are utilized.

3.1 Measuring Gaming the System

We leverage the model of gaming developed by [14] to annotate transactions as gamed. This model identifies a set of patterns of transactions that experts identify as gamed patterns. A student is determined to be gaming at some time if a series of transactions matches an identified transaction. For instance, a common pattern is when students enter the same or a very similar answer into multiple places without answering correctly, effectively guessing where a calculation result belongs without understanding the organization of the problem. Another common pattern is when students ask for help without taking much time to consider the problem, followed shortly after by an incorrect input. In this case, the student appears to be using the help facility to get an answer but is not taking enough time to use the information provided to derive an answer. The dataset consists of 4.1% of transactions as being labeled as part of a gaming behavior, where the majority of students are labeled as gaming between 3.2 to 4.3% of all observed transactions.

3.2 Aligning Session Time

The data described above only includes transactions after eliminating certain transactions from the original dataset. In order to see temporal patterns, data was excluded from short sessions with length in the bottom 5% of all student session lengths, which was determined to be about 5 minutes. The resulting observed student sessions ranged from 5 minutes to 58 minutes, with a median length of 32 minutes.

One difficulty in measuring ego-depletion with observational data is in controlling for differences in the depleting effects of context. In ego-depletion studies, the task is controlled for and thus can be ruled out to explain observed differences in behavior. In intelligent tutoring contexts. The adaptive instruction will provide variably challenging and types of content and may differentially deplete students across the experiences within the same period of time. To overcome this issue, we leverage the insight that when two students begin working, they might be in similar states relative to their internal thresholds for self-regulation. We also assume that when two students stop working, they are in comparable states. If these two students stop working at different times, it implies similar start and finish attention states, but different depleting effects of context that were experienced over time. In order to account for these differences in uncontrolled contextual factors, we created an additional time measure that aligned individual student transactions within sessions by the percentage of the session time that has elapsed. This alignment facilitates comparison of transactions relative to the start and end of a session, scaled to the session length.

4. Modeling the Effect of Time

Theories of self-regulation imply different models of the effect of time on self-regulation. Attentional shift models posit a cognitive cost of task switching. These costs may cause some tasks to seem more difficult near the beginning of a session. Ego-depletion models imply a reduction of a limited capacity to self-regulation resource over time. These models suggest students may eventually find it difficult to continue in a task and signs of fatigue, such as gaming, may be revealed by an increased tendency to engage in gaming behaviors before finishing working. To test these model implications, we compare five random effect logistic regression models to determine how self-regulation may vary over the course of a session.

We introduce M1 as the baseline model for comparison to determine if any temporal models are significantly more predictive than current best practices as suggested by prior gaming research. This model includes random effects for both student and curricular section to control for the previously established impacts of student and context on student's tendency to game. The remaining four subsequent models similarly control for student and contextual factors while introducing additional factors representing temporal effects.

To define the remaining four models, time is represented along two dimensions. In the first dimension, time is represented as either time elapsed since the student began working or percentage of total working time elapsed, as described section 3.2. Time elapsed models represent the default model informed by both ego-depletion and task switching theories. Percentage of time elapsed models test the hypothesis that such a representation better captures motivation as temporally relative to the most informative moments of student behavior. In the second dimension, time is represented linearly or quadratically. Linear models allow only one main temporal effect to be captured by the model, either a constant increase or decrease in motivation over the course of a session. Quadratic models can capture different effects at the start and end of the session that differ from each other and the middle of the session. All temporal variables are normalized over the full dataset for model interpretation.

M4.1: Baseline – Baseline model for comparison controlling for differences in student's tendency to game and contextual factors across curricular sections, such as average difficulty, that influence gaming.

$$Eq\ 4.1: Gaming \sim (1|Student) + (1|Section)$$

M4.2: Linear Session Time – Extending the baseline model M4.1 by adding a linear term for time-elapsed since the student has begun working

$$Eq\ 4.2: Gaming \sim time-elapsed + M4.1$$

M4.3: Linear Percent Time – Extending the baseline model M4.1 by adding a linear term for proportion of session time elapsed as a percentage of total time observed working.

$$Eq\ 4.0: Gaming \sim pct-time-elapsed + M4.1$$

M4.4: Quadratic Session Time – This model extends model M4.2 by adding a quadratic term

$$Eq\ 4.0: Gaming \sim time-elapsed^2 + M4.2$$

M4.5: Quadratic Percent Session Time – In addition to the random effects in Eq 4.1, this model tests the hypothesis that students self-regulation resources are

$$Eq\ 4.0: Gaming \sim pct-time-elapsed^2 + M4.3$$

4.1 Comparing Models

Table 1. Comparing models of temporal trajectories of student gaming behaviors

Model	BIC	AIC	LogLik
M4.1	434741	434703	-217348
M4.2	434682	434632	-217312
M4.3	434668	434619	-217305
M4.4	434454	434392	-217191
M4.5	454503	434441	-217215

The results of fitting each of the five models are shown in Table 1, including model performance as assessed by AIC, BIC, and log-likelihood. In general, all models with temporal factors outperform the baseline model, M4.1. This implies that temporal information has a significant effect on student's self-regulation behaviors. Additionally, both quadratic models, M4.4 and M4.5, are significantly better than their linear counterparts (Chisq = 179 ($p < 0.001$) for M4.2 vs M4.4, and Chisq = 242 ($p < 0.001$) for M4.3 vs M4.5). Likewise, M4.4 and M4.5 are significantly better than baseline with Chisq = 315 ($p < 0.001$) and Chisq = 266 ($p < 0.001$) respectively.

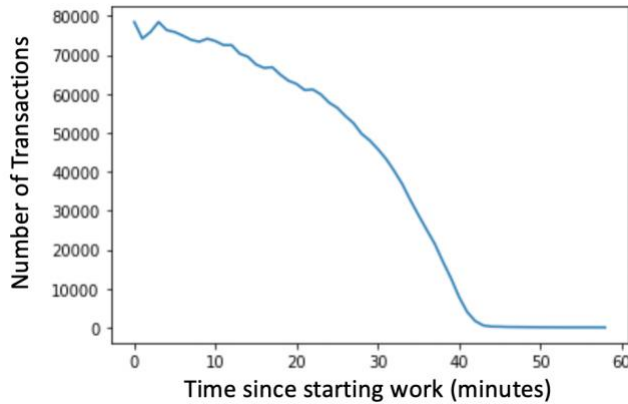


Figure 1. Number of observations over time in session

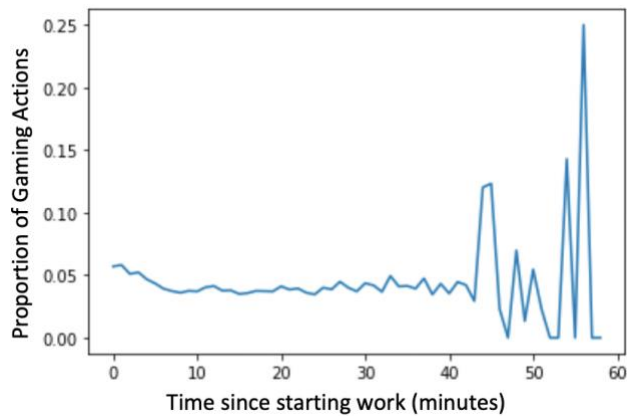


Figure 2. Proportion of Gaming Actions by minute

Exploratory plots of proportion of gaming the system transactions over the session support these interpretations. Figure 2 and 4 plot the proportion of transactions identified as gaming the system behaviors across the session over minutes passed or proportion of total session time respectively. As expected from the quadratic fit models, each figure shows an increased proportion of gaming behaviors near the start and end of sessions.

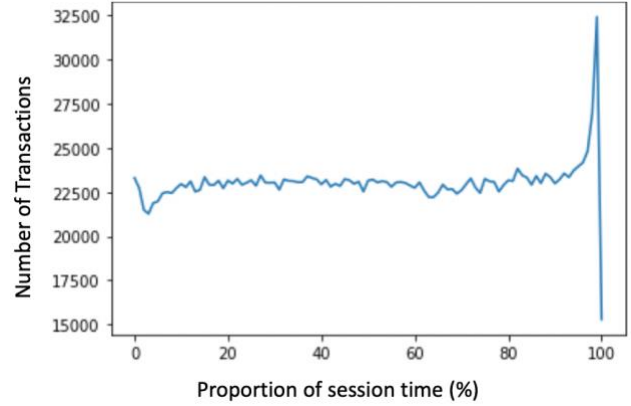


Figure 3. Number of observations over proportion of session time

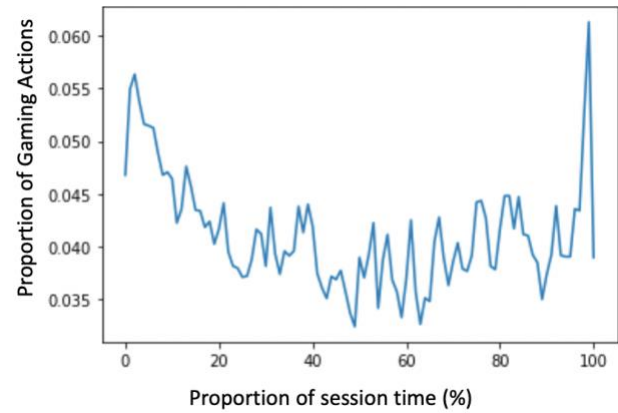


Figure 4. Proportion of Gaming Actions by Proportion of Session Time Passed

A closer look at the data in Figure 1 reveals that there is a large student participation drop-off near the 43 minute mark. While whole class sessions seem to regularly measure about 60 minutes, students' login and logout times are quite staggered such that 99% of observed student sessions are less than 43 minutes in length. Only 82 out of more than 9800 sessions are observed where students worked continuously for between 43 and 60 minutes. Furthermore, analyzing gaming averaged over each minute of the hour, Figure 2, shows that this dramatic reduction in data is associated with very large and volatile estimates of average students gaming per unit time. Because of the low amount of data observed in the last 17 minutes of sessions longer than 43 minutes, it is hard to draw stronger conclusions about whether students are much more likely to display gaming behaviors if they are able to stay on task longer than 43 minutes, or if the volatility is due to random sampling bias.

A closer inspection of data in Figure 3 also shows some peculiar variability in data at the start and end of sessions. Because session time is divided evenly across the proportion of sessions, there is

no a-priori reason to believe students have more or less frequent transactions at any time in the session. The small decrease in quantity of transactions near the start of sessions implies students take longer on average to complete actions near the start of work. The large spike of activity near the end implies students are taking less time per action shortly before stopping work. In both cases, the data sparsity issue seen in Figure 1 is not likely driving the changes in proportion of gaming seen in Figure 4. The small decrease in activity near the start is associated with the start of a broader downward trend in proportion of gaming behaviors that continues even after activity frequency flattens. The sudden increased frequency of transactions near the end of sessions is associated with a comparable spike in prevalence of gaming the system behaviors. However, because some gaming behaviors are defined by rapid actions in succession, this relationship is expected.

Taking the model comparisons and exploratory data analysis together, this evidence supports the interpretation that there are non-monotonic differences in gaming the system behaviors between the start, middle, and end of sessions.

Table 2. Model coefficients for M4.4 and M4.5

Term	M4.4 - β	Term	M4.5 - β
Intercept	-4.215	Intercept	-4.217
Percent time elapsed	-0.265	Time elapsed	-0.283
(Percent time elapsed) ²	0.231	(Time elapsed) ²	0.252

Comparing the two quadratic models, M4.4 is the best fit model by all 3 measures, BIC, AIC, and Log Likelihood. The model details can be seen in Table 2. The variance in gaming attributable to curricular sections is 0.87. This translates to average gaming attributable to tutor context level factors to range between 0.23% and 8.4% for 95% of sections. The variance attributable to students is much smaller, 0.088. This translates to average gaming attributable to trait-level student factors to range between 0.82% to 2.57%. An inspection of the model coefficients shows that the model predicts the average gaming level at the start of a session, $P(\text{gaming}|t=0)$, is 4.1%. Average gaming at the end of the session, $P(\text{gaming}|t=60 \text{ minute})$, is 18.7%. The quadratic model reaches a minimum observed gaming of 1.3% at 23 minutes into the session.

An 18.7% average probability of gaming after working for 60 minutes appears to be very high given that gaming only occurs overall in the dataset in about 4.5% of all actions. As discussed in the previous exploratory data analysis, the very high gaming proportion observed in the last 17 minutes of sessions is potentially related to the increased volatility created from estimates drawn from small amounts of data. These estimates spike upwards as high as 25%, which corresponds with the dramatic difference between start and end gaming predicted by M4.4. Therefore, the model is reflecting this same artifact of the data.

Inspecting M4.5, the model predicts that gaming is more likely in the start and end of the session. The average probability of gaming decreases to 1.35% by the time the student has worked 67% of the total time. According to the model, we are 3.34 times more likely to observe students game the system near the start of work than near their peak level of focus. Likewise, it is 1.32 times more

likely to observe gaming the system in the moments shortly before students stop work. This model appears to make less dramatic predictions that are more inline with expectations based on overall average frequencies of gaming while not reflecting the same uncertainties as M4.4.

These results support the hypothesis that self-regulation processes have an impact on the average occurrence of gaming the system behaviors over the course of a work session. Students in this data appear to experience decreased motivation near the start of work as would be predicted by the cognitive costs of task switching. Likewise, students appear to show some decreased motivation before stopping work as predicted by ego-depletion theories.

5. Leveraging Gaming for Prediction

The previous analysis has demonstrated that observing instances of weaker self-regulation, such as gaming the system behaviors, support a view of student's dynamic self-regulation capacities over time as predicted by ego-depletion and task-switching theories. This raises the natural question of exactly what observing such lapses in self-regulation implies about a student's internal capacities.

5.1 Gaming Indicates Cognitive Effort

If students are not observed to game the system early in a session, we expect that student motivation is likely higher around this time despite the brief slightly negative impact of task switching. This greater motivation allows students to bring greater cognitive resources to the work relative to days when gaming is observed near the start. When comparing assistance rates in the beginning of a session, the proportion of questions either answered incorrectly or with a request for help on first attempt, a student who is more cognitively engaged should be less likely to make errors or ask for help. Likewise, similar patterns should be associated with assistance rates near the end of students work.

We compared the assistance rates for sessions where a student is observed gaming in the first 10% of the session time (the first 3 minutes for the median session) to assistance rates where no gaming is observed in the first 10% of the session time. To calculate the assistance rate, the raw student transactions are aggregated by problem-step. The outcome of each step is determined by the first attempt at the step. The step is labeled as gaming the system if any of the aggregated transactions are labeled as gaming. Because patterns of gaming generally involve either incorrect or help-seeking behaviors, steps that were labeled as gaming the system are removed before calculating the proportion of incorrect and help-request steps to overall steps observed in the portion of the session.

The assistance rates in the start of sessions are shown in Figure 5 and were found to be significantly lower ($t=-15.22$, $p < 0.001$). The average assistance rate where gaming is observed is 30% ($sd=25$) while the average rate when gaming is not observed is 21% ($sd=26$). Similarly, Figure 6 shows boxplots for assistance rates in the last 10% of sessions. Rates were found to be to be significantly lower ($t=-11.6$, $p<0.001$) with the average session where gaming is observed having a rate of 25.3% ($sd=22$) compared to the average non-gaming session having a rate of 18.6% ($sd=24$).

This simple analysis does not take into account factors such as question difficulty. It is possible that if students are working on difficult content near the start, then they are more likely to make errors and request hints. It also implies that more challenging material may impact how students evaluate the likelihood of

prospective reward given their perceived abilities. This may lead students to believe that applying effort is unlikely to result in experiencing the reward or attempting to apply effort may have greater depleting effects that impact future actions. In either case, it is possible that more challenging material instead of task-switching or ego-depletion explains the relationship between increased assistance score and gaming behaviors near the start and end of work. However, these tests do provide compelling evidence for a possible impact of decreased cognitive engagement on some practice opportunities that can inform future modeling work.

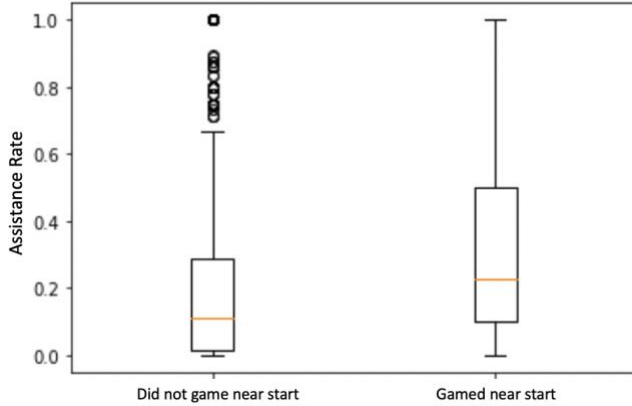


Figure 5. Comparing assistance rate at the start of sessions

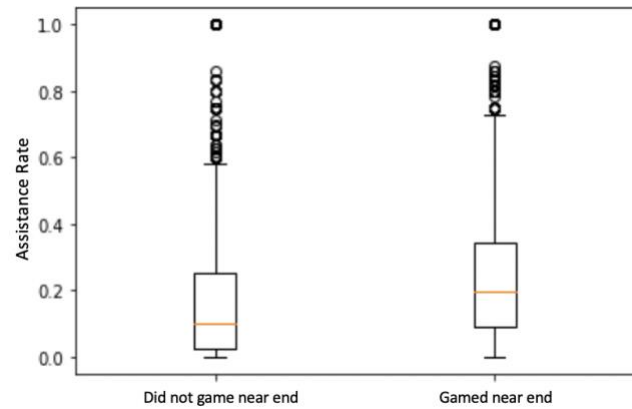


Figure 6. Comparing assistance rate at the end of sessions

5.2 Gaming Indicates Motivation Levels

Student's day-to-day average motivation level is affected by factors in the school, in the classroom, and in the student's life more broadly. A death in the family, a fight with a significant other, or a poor grade in another class might be weighing on a student's mind while that begin working. These factors may have a negative effect on student's ability to self-regulate throughout the entire session. If this is the case, these factors will act in combination with the additional impacts of task-switching or ego-depletion at the start and end of the session to impact a student's capacity to self-regulate. Thus, observing gaming the system behaviors at the start or end of a session may also be informative about a student's more general motivational level. In this section, we analyze gaming behaviors throughout the session using information about whether students gamed at the beginning or end of a session to improve predictions of gaming in the rest of the session.

Gaming at the start and end are defined the same as in the previous section. In the data, 29.7% of sessions are observed with gaming at the start while 32.0% of sessions have gaming at the end. Together 49.9% of sessions have instances of gaming the system in the start or end, while only 11.8% of sessions are observed with gaming in the start and end of the session. While gaming near the start or end might be indicative of session level motivational impacts, in this analysis we test whether seeing any gaming at the start or end is sufficiently informative or if start and end are differently informative.

To perform this analysis, we use the best model from the Section 4 analysis, M4.4 the quadratic percent-time-elapsed model. This model will control for the variance due to student and tutor contextual factors, removing concerns about confounds such as gaming at the start may be due to generally more difficult material that makes gaming more likely throughout the session. We compare models that add main effects for whether gaming was observed at the start or at the end as well as linear and quadratic interaction effects. The models are elaborated as follows:

M5.1: Baseline Quadratic Model – the baseline model from Section 4 analysis for comparison.

Eq 5.1: $\text{Gaming} \sim \text{pct_elapsed} + \text{pct_elapsed}^2 + (1|\text{Stu}) + (1|\text{Sect})$

M5.2: Gaming at start/end main effect – M5.1 with a binary indicator variable of whether gaming is observed near the beginning of the session and a binary indicator variable of whether gaming is observed near the end of the session

Eq 5.2: $\text{Gaming} \sim \text{M5.1} + \text{g_start} + \text{g_end}$

M5.3: Combined Gaming at start or end main effect – M5.1 with a binary indicator of whether gaming is observed at either the beginning or the end of the session

Eq 5.3: $\text{Gaming} \sim \text{M5.1} + \text{g_start_end}$

M5.4: Gaming at start and end with linear interactions – M5.4 elaborates on top of M5.2 adding linear interactions with time.

Eq 5.4: $\text{Gaming} \sim \text{M5.2} + \text{g_start:pct_elapsed} + \text{g_end:pct_elapsed}$

M5.5: Gaming at start and end with quadratic interactions – M5 elaborates on top of M5.4 adding interactions with quadratic time terms.

Eq 5.5: $\text{Gaming} \sim \text{M5.4} + \text{g_start:pct_elapsed}^2 + \text{g_end:pct_elapsed}^2$

Comparing M5.2 and M5.3, we see that including separate main effects for gaming at the start and gaming at the end leads to better models rather than combining the information into a single indicator of whether there were any self-regulation failures at either the start or the end of the session. This particular result is worth further investigation to understand how and why self-regulation at the start of a session is differently indicative of student motivation levels compared to gaming at the end of the session.

The results in Table 3 indicate the best fit model is M5.5, the model with start/end gaming information and interactions with linear and quadratic terms. This model is significantly different from the baseline quadratic model ($\text{Chisq}=49.42, p<0.001$) and establishes the informativeness of gaming in the start or end of a session on student's motivation levels through the time that students are working. Details about the model are given in table 4.

Table 3. Comparing Gaming Predictions using Start/End Gaming

Model	AIC	BIC	LogLik
M5.1	434441	434503	-217295
M5.2	422316	422403	-211151
M5.3	427322	427397	-213655
M5.4	419913	420045	-209958
M5.5	418266	418402	-209122

The variance accounted for by section and student level random effects are reduced in comparison to the baseline quadratic model reported in Section 4. The variance attributable to student factors was found to be 0.0789, which translates to an average gaming level of 0.64% to 1.91% for 95% of students. The variance attributable to section level factors was found to be 0.7527, which translates to an average gaming frequency of 0.20% to 5.79% for 95% of sections. This implies that a significant fraction of observations of gaming that were previously explained by section-level factors appears to now be explained by motivational factors indicated by gaming at the start or end of a session.

Table 4: Coefficients for start/end gaming with quadratic interaction terms

Term	β
Intercept	-4.489
Percent time elapsed	1.129
(Percent time elapsed) ²	-1.251
Gamed at start	0.301
(Gamed at start) * Percent time elapsed	-1.480
(Gamed at start) * Percent time elapsed) ²	1.170
Gamed at end	0.356
(Gamed at end) * Percent time elapsed	-0.490
(Gamed at end) * Percent time elapsed) ²	0.900

Table 5 contains the predicted gaming attributable to the main effect terms in model M5.5. The first column describes average predicted gaming at the start of work. The third column describe average predicted gaming at the end of work. Because the model includes quadratic terms, the second column is included to describe the optimum (minimum or maximum) probability of gaming throughout the session. The fourth column describes the odds ratio the chance of gaming at the start relative to the optimum point. The fifth column describes the odds ratio of the chance of gaming at the end compared to gaming at the optimum point. The complexity of the model can make it challenging to interpret, however there are some important trends indicated by the model. If gaming is observed only in the start of a session, gaming is most likely to occur similarly near the start and will reduce over the course of the session as evidenced by the odds of gaming being greatest at the start relative to the end. Likewise, observing gaming only at the end of the session implies that students tend to be well regulated near the beginning of the session and will appear to fatigue over the session until near the end where the odds fall slightly. When students are not observed

gaming at the start or end, there is a corresponding low probability of observing gaming near the start and end. However, over the course of the session, the model predicts that these students become more likely to have slightly reduced motivation until the latter half of the session where attention on the time pressure of the end of class might increase motivation through the end of class. In the limited sessions where students are observed gaming at the start and end, the model predicts a much greater propensity to game throughout, with a 53% chance in the start and a 5% chance near the end.

Table 5: P(Gaming) Main effect predictions given start/end gaming observations

Context	Game (t=0)	Game (t=opt)	Game (t=100)	Start Odds	End Odds
No Gaming start or end	0.35%	1.43%	0.21%	0.24	0.15
Start Gaming	2.14%	2.14%	0.66%	1	0.31
End Gaming	0.18%	2.10%	1.71%	0.086	0.81
Start + End Gaming	53.1%	1.72%	5.1%	30.9	2.98

Taken together, these results support the conclusion that gaming at the start and end of work are indicative of session-level motivational factors influencing student behavior. It also provides initial evidence for separable constructs indicated by gaming at the start versus at the end. Each of these constructs appears to have different degrees of impact on underlying student motivation factors and the resulting decision processes that lead to observable behaviors.

6. Discussion

We have treated gaming the system behaviors as indicators of student's self-regulation. Task switching and ego-depletion theories of self-regulation predict a temporal pattern to student's abilities to self-regulate over the course of a class period. Predictive model comparisons are supportive of the hypothesis that both task switching and ego-depletion are evident in the patterns of student behaviors over each class session. Further analysis indicates that observations of self-regulation behaviors in the start and end of class might be indicative of both temporally immediate degrees of cognitive engagement as well as more session or day-level influences on motivation.

Open questions remain about how student models could operationalize task switching or ego-depletion. The work presented, uses information about the full student session to represent time, though such information is not available to real-time models. This raises the question of how should student's prior behaviors inform a predictive models of student ability to task switch or ego deplete? To what degree do students display consistency in their ability to task switch quickly or manage ego-depletion more effectively across sessions? Over the course of months or years? To what degree are these capacities independent or can correlations be attributable to other latent motivational causes?

We believe these findings highlight the importance of leveraging student models that incorporate temporal variables in the design

of learning activities. Problem selection algorithms may want to be biased for lower challenge or greater interest to overcome negative effects of task switching. Similarly, activities may want to incorporate changes in the rhythm of the activity in order to periodically re-engage student attention as it wains over time. This work exposes an unexplored design space for how educational activities could incorporate temporal effects of student motivation to better enable student learning.

In this work, we introduce the importance of considering temporal factors in addition to content-related cognitive factors to more effectively support students' motivational trajectories within a work session. These findings extend the rich body of work on modeling student motivational and cognitive processes with self-regulated learning. Students are not machines, and they do not always jump immediately into tasks full throttle or have the endurance to work as long as they are asked. Hopefully, a future that recognizes these dynamics can take intelligent tutoring systems one step closer to emulating the capabilities of effective teachers.

7. ACKNOWLEDGMENTS

The research reported here was supported, in whole or in part, by the Institute of Education Sciences, U.S. Department of Education, through grant R305B150008 to Carnegie Mellon University. The opinions expressed are those of the authors and do not represent the views of the Institute or the U.S. Department of Education.

8. REFERENCES

- [1] Bernacki, M.L. and Ritter, S. Dataset 613 in DataShop. Retrieved from <https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=613>.
- [2] Botvinick, M. and Braver, T. 2015. Motivation and Cognitive Control: From Behavior to Neural Mechanism. *Annual Review of Psychology*. 66, 1 (Jan. 2015), 83–113.
- [3] Cho, K. et al. 2015. Modeling how incoming knowledge, persistence, affective states, and in-game progress influence student learning from an educational game. *Computers & Education*. 86, c (Aug. 2015), 224–235.
- [4] Dumdumaya, C.E. et al. 2018. Identifying Students' Persistence Profiles in Problem Solving Task. In Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP'18). 281–286.
- [5] Eccles, J.S. 2005. Subjective task value and the Eccles et al. model of achievement-related choices. *Handbook of Competence and Motivation*. 105–121.
- [6] Flake, J.K. et al. 2015. Measuring cost: The forgotten component of expectancy-value theory. *Contemporary Educational Psychology*. 41, C (Apr. 2015), 232–244.
- [7] Karumbaiah, S. and Shute, V. 2018. Predicting Quitting in Students Playing a Learning Game. In *Proceedings of the 11th International Conference on Educational Data Mining* (Jul. 2018).
- [8] Kool, W. et al. Decision making and the avoidance of cognitive demand. *Journal of Experimental Psychology: General*. 139, 4, 665–682.
- [9] Marzano, R.J. et al. 2003. Classroom management that works: Research-based strategies for every teacher.
- [10] Mills, C. et al. 2014. To Quit or Not to Quit: Predicting Future Behavioral Disengagement from Reading Patterns. In *Proceedings of the International Conference on Intelligent Tutoring Systems*. Springer, Cham. 19–28.
- [11] Nieuwenhuis, S. and Monsell, S. 2002. Residual costs in task switching: Testing the failure-to-engage hypothesis. *Psychonomic Bulletin Review*. 9, 1 (Mar. 2002), 86–92.
- [12] Paquette, L. and Baker, R.S. 2017. Variations of Gaming Behaviors Across Populations of Students and Across Learning Environments. In *Proceedings of the 18th International Conference on Artificial Intelligence in Education* (Cham, Jun. 2017), 274–286.
- [13] Paquette, L. et al. 2015. Cross-System Transfer of Machine Learned and Knowledge Engineered Models of Gaming the System. In *Proceedings for the International Conference on User Modeling, Adaptation, and Personalization* (Apr. 2015), 183–194.
- [14] Paquette, L. and de Carvalho, A.M.J.A. 2014. Towards Understanding Expert Coding of Student Disengagement in Online Learning. In *Proceedings for the Annual Meeting of the Cognitive Science Society* (2014), 1126–1131.
- [15] Scher, S.J. and Osterman, N.M. 2002. Procrastination, conscientiousness, anxiety, and goals : Exploring the measurement and correlates of procrastination among school-aged children. *Psychology in the Schools*. 39, 4 (May 2002), 385–398.
- [16] Steel, P. 2007. The nature of procrastination: A meta-analytic and theoretical review of quintessential self-regulatory failure. *Psychological bulletin*. (2007), 65–94.
- [17] Stewart, A. et al. 2017. Face Forward: Detecting Mind Wandering from Video During Narrative Film Comprehension. In *International Conference on Artificial Intelligence in Education* (Cham, Jun. 2017), 359–370.
- [18] Stewart, A. et al. 2016. Where's Your Mind At?: Video-Based Mind Wandering Detection During Film Viewing. In *Proceedings for the 2016 Conference on User Modeling, Adaptation, and Personalization* (New York, New York, USA, Jul. 2016), 295–296.
- [19] Wigfield, A. and Eccles, J.S. 2000. Expectancy–Value Theory of Achievement Motivation. *Contemporary Educational Psychology*. 25, 1 (Jan. 2000), 68–81.
- [20] Wolters, C.A. 2004. Advancing Achievement Goal Theory: Using Goal Structures and Goal Orientations to Predict Students' Motivation, Cognition, and Achievement. *Journal of Educational Psychology*. 96, 2 (2004), 236–250.
- [21] Wolters, C.A. 2003. Understanding procrastination from a self-regulated learning perspective. *Journal of Educational Psychology*. 1, 95 (2003), 179.
- [22] Muldner, K. et al. 2011. An analysis of students' gaming behaviors in an intelligent tutoring system: predictors and impacts. In *Proceedings for the Conference on User Modeling and User-Adapted Interaction* (Jan. 2011), 99–135.
- [23] Schmeichel, B.J. 2007. Attention control, memory updating, and emotion regulation temporarily reduce the capacity for executive control. *Journal of Experimental Psychology General*. 136, 2 (2007), 241–255.
- [24] Baker, R.S. et al. 2004. Detecting Student Misuse of Intelligent Tutoring Systems. In *Proceedings of the International Conference on Intelligent Tutoring Systems* (Berlin, Heidelberg, Aug. 2004), 531–540.
- [25] Baker, R.S. 2007. Is gaming the system state-or-trait? Educational data mining through the multi-contextual application of a validated behavioral model. In *Complete On-Line Proceedings of the Workshop on Data Mining for User Modeling at the 11th International Conference on User Modeling* (2007).
- [26] Baker, R.S. et al. 2004. Off-task behavior in the cognitive tutor classroom: when students game the system. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, New York, USA, 2004), 383–390.
- [27] McClelland, M.M. and Cameron, C.E. 2011. Self-Regulation in Early Childhood: Improving Conceptual Clarity and Developing Ecologically Valid Measures. *Child Development Perspectives*. 6, 2 (Jul. 2011), 136–142.

Can we Take Advantage of Time-Interval Pattern Mining to Model Students Activity?

Oriane Dermay, Armelle Brun
Université de Lorraine, CNRS, Loria
Campus Scientifique
54506 Vandœuvre-lès-Nancy, France
name.surname@loria.fr

ABSTRACT

Analyzing students' activities in their learning process is an issue that has received significant attention in the educational data mining research field. Many approaches have been proposed, including the popular sequential pattern mining. However, the vast majority of the works do not focus on the time of occurrence of the events within the activities. This paper relies on the hypothesis that we can get a better understanding of students' activities, as well as design more accurate models, if time is considered. With this in mind, we propose to study time-interval patterns.

To highlight the benefits of managing time, we analyze the data collected about 113 first-year university students interacting with their LMS. Experiments reveal that frequent time-interval patterns are actually identified, which means that some students' activities are regulated not only by the order of learning resources but also by time. In addition, the experiments emphasize that the sets of intervals highly influence the patterns mined and that the set of intervals that represents the human natural time (minute, hour, day, etc.) seems to be the most appropriate one to represent time gap between resources.

Finally, we show that time-interval pattern mining brings additional information compared to sequential pattern mining. Indeed, not only the view of students' possible future activities is less uncertain (in terms of learning resources and their temporal gap) but also, as soon as two students differ in their time-intervals, this difference indicates that their following activities are likely to diverge.

Keywords

Students behavioral patterns, time-interval pattern mining, interval granularities, sequential pattern mining.

1. INTRODUCTION

The wealth of data that can be collected from a Learning Management System (LMS), mainly the logs of students' interactions with learning resources, provide opportunities to

get a more comprehensive understanding of students learning process: point out engaged or at-risk students, identify the most commonly studied or the most difficult resources, highlight recurrent students' activities, etc. In addition to this thorough understanding, inferences or decisions can be drawn: estimate students outcome, predict students future behavior (including dropout), personalize learning by providing students with information or recommendations, etc. To carry out such understanding, inference or decision, data mining methods have been applied. Pattern mining, that discovers frequent patterns of events in data, is one of these methods and is also used in a large number of application fields. Sequential Pattern Mining (SPM) consists of discovering patterns when data is sequential in nature. These patterns, named sequential patterns, are frequent ordered sequences of events.

In the educational field, a sequential pattern often represents a recurrent sequence of learning resources, that we call an activity [30, 5].

The time of occurrence of events is often part of the data to be mined. However, in most of the cases, the patterns mined do not contain temporal information. Nevertheless, the literature has introduced different ways of including such information in patterns. We can, for example, cite temporal patterns, made of events that are associated with their time of occurrence [36], their duration [8], or the time gap between the events. In [9], gaps between events are grouped into intervals, resulting in time-interval sequential patterns. Since a time-interval pattern conveys more information than its corresponding sequential pattern, they are still the focus of research works [33]. In the rest of the paper, time-interval patterns will be referred to as *ti*-patterns and sequential patterns to as *s*-patterns.

We think that *ti*-patterns are adequate to represent students' activities. Indeed, it is rare that two students perform exactly the same activities, in both learning resources and time, even though they share underlying sequential activities. To the best of our knowledge, no work in the field of educational data mining has focused on the mining of *ti*-patterns.

In this work, we thus rely on the hypothesis that mining *ti*-patterns will contribute to a better view and understanding of students' learning activities. These patterns do not only indicate in which order students interact with learning resources, but provide also information about the temporal relationship between these resources. For example, let us

Oriane Dermay and Armelle Brun "Can we Take Advantage of Time-Interval Pattern Mining to Model Students Activity?" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 69 - 80

consider that students tend to interact sequentially with two resources, each of them being lecture slides. The sequence of both resources represents a sequential activity. Suppose that mining ti -patterns highlights that the time gap between both resources tends to be less than 1 minute for some students and between 2 and 4 hours for others. We can thus deduce beyond this sequential activity that there are two typical behaviors.

To support our hypothesis, we will conduct a study to evaluate if ti -patterns can be actually identified from students' activity data and evaluate to what extent ti -patterns provide additional information about students' activities.

In the following sections, we will first present an overview of related works on sequential and temporal pattern mining (Section 2). We then present the methodology we adopt to support the hypothesis that we draw (Section 3). Section 4 details the experiments we conduct on a real dataset and presents some ti -patterns. The last sections discuss the results (Section 5), then conclude the work and present our expected future work (Section 6).

2. LITERATURE REVIEW

2.1 Sequential Pattern Mining (SPM)

Sequential Pattern Mining is a popular task in Data Mining, introduced by Agrawal and Srikant in [1]. SPM aims to discover frequent sequential patterns in sequential databases. A sequential database D is a set of tuples $D = \{(sid_i, d_i)\}$, where sid_i is the unique identifier of a sequence, and d_i an input sequence. A sequence is an ordered list of events: $s = \langle E_1 E_2 \dots E_x \rangle$, with $E_i \in E$ the set of events. To understand what a frequent sequential pattern is, let us first define what a sub-sequence is. $\alpha = \langle E_1 \dots E_n \rangle$ is a sub-sequence of $\beta = \langle E'_1 \dots E'_m \rangle$ if:

$$\exists [1 \leq j_1 \leq \dots \leq j_n \leq m] \{E_1 = E'_{j_1}, \dots, E_n = E'_{j_n}\}.$$

We also say that β is a super-sequence of α , or that it contains α . Let us now define what the support of a sequence is. The support of α , noted $supp(\alpha)$, is the number of sequences in the sequential database D that contain α . Based on both definitions, we can now define that a sub-sequence α is a frequent sequential pattern, if $supp(\alpha) \geq \delta$, for a defined minimum support threshold δ . We define SP as the set of frequent sequential patterns.

Many SPM algorithms have been proposed in the literature. The most commonly cited ones are *GSP* [32], *PrefixSpan* [26], *SPADE* [37]. All these algorithms use the "apriori property": "If a sequence s is not frequent, then none of the super-sequences of s is frequent." Thus, when one pattern is infrequent, it is not extended. Algorithms can be divided into two main approaches. Apriori-like algorithms (also called *breadth-first* search algorithms), such as *Generalized Sequential Pattern Mining* (GSP) algorithm [32], are the first algorithms that have been proposed. However, these algorithms suffer from scalability problems, mainly due to memory requirements. *Depth-first* search algorithms, which include pattern-growth algorithms, do not suffer from memory complexity, which explains their popularity.

For a couple of years, the most common SPM algorithm is the *Prefix-Projected Sequential Pattern Growth* (*PrefixSpan*) algorithm [26], which is a pattern-growth algorithm, that

relies on projected databases. Projected databases generally reduce the research space as the size of the projected databases decreases at each iteration. However, the main cost is linked to the generation of these projected databases. The pseudo-code of *PrefixSpan* is presented in Algorithm 1.

Algorithm 1 PrefixSpan (α, l, D)

```

1: Inputs:
2:  $\alpha$ : a sequential pattern and  $l$  its length.
3:  $D$ : a sequential database, or a projected database.
4: Outputs:
5:  $SP$ : the set of all frequent sequential patterns.
6: Method:
7: Scan  $S$  to find all frequent items  $b$ .
8: for all  $b$  do
9:   add  $\alpha' = \langle \alpha b \rangle$  to  $SP$  as a new sequential pattern.
10: end for
11: for all  $\alpha'$  do
12:   create the  $\alpha'$ -projected database  $D|_{\alpha'}$ 
13:   call PrefixSpan( $\alpha', l + 1, D|_{\alpha'}$ )
14: end for

```

In the works mentioned below, both the database and the patterns are sequential. However, in some cases, the database can be temporal, i.e. contain information about the time of occurrence of the events. In these cases a sequence is defined as: $s = \langle (t_1, E_1), (t_2, E_2), \dots, (t_n, E_n) \rangle$. where (t_i, E_i) represents an event E_i and its time of occurrence t_i .

When sequential patterns are mined from these databases, time can be either used as an information or order between events, such as in SPADE [37]. The time of appearance of events can also be used as a constraint. For example, in [18] the authors consider that when two consecutive items in a sequence are separated by a time gap bigger than a predefined threshold, they are temporally too distant to represent an association that makes sense. In the same context, [31] discards uninteresting patterns by introducing an interval constraint between items.

2.2 Sequential Patterns Mining in EDM

Sequential Pattern Mining has been extensively used in Educational Data Mining. They are mainly used to identify frequent patterns of students' activities [16, 28], including those that maximize the student learning performance [10]. In [21] SPM is used to study the differences in students' productive and unproductive learning behaviors and thus identify high versus low performing students. A similar objective has been studied on group work systems to understand the success factors in groups behavior [27, 25].

SPM is also used to detect learning problems early, such as in [20] where frequent sequential patterns and flag interaction sequences that are indicative of problems are mined.

One step further, SPM can act as a first step in decision making. In [7], the prerequisite structure of skills is found out, by identifying relations between variables from data. The algorithms developed in [28, 34, 11] provide students with personalized recommendations of learning resources according to their current activity or their learning style. A complete view of various approaches used in educational data mining is presented in [2].

2.3 Temporal Pattern Mining

Temporal information appears to be fundamental in many contexts, hence the number of works interested in the mining of patterns that contain temporal information. Here again, time information can be used in several ways and for different goals: gaps, duration, intervals, etc.

Time information is often used as a gap between events of a pattern. For example in [36], the author considers that each occurrence of a sequential pattern may have different, but close, temporal elements. So, they propose to associate each pair of events of a pattern with a minimal, a mean and a maximal gap values between these events. The resulting model is made of sequential patterns enriched with temporal information, called delta patterns. Similarly, [15] proposes to add temporal information to each pair of events in a sequential pattern. This information, referred to as an annotation, represents a typical gap value between each pair of events of a pattern. In this work, the acceptance of the variation around this typical gap value is automatically evaluated. At the opposite of the previous works, [35] pre-defines a maximal gap value between events of a pattern, which results in temporal patterns called chronicles. [22] introduces an even more constraining frame, the exact gap interval value is imposed. This approach results in a decrease in the support of each pattern. Thus, the number of extracted patterns decreases.

In addition to the gap value, [17] exploits the duration of events. Each element of a pattern is composed of the event, associated with its begin and end timestamps. They propose an Apriori-like algorithm, that uses a hypercube representation of temporal sequences.

More recently, [13] introduces an Apriori-like temporal pattern mining algorithm on multi-modal data streams. At the opposite of the previous works, they do not only use the time gap between events (that represents the duration of the event), but also use the exact starting time of each event.

In line with the works presented above, [6] also manages gap values between events, that are grouped into intervals. At the opposite of other works, the intervals of gap values are predefined, and form "time-interval sequential patterns". A time-interval sequence is defined as:

$$\alpha = \langle E_1\tau_1 E_2\tau_2 \dots \tau_{l-1}E_l \rangle$$

where $E_i \in E$ is the set of events for $1 \leq i \leq l$ and $\tau_i \in TI$ the set of time-intervals. The sequence α is a time-interval pattern if $\text{supp}(\alpha) \geq \delta$. We note TP the set of frequent time-interval patterns of a database D . In their article, the authors propose two algorithms called *I-Apriori* and *I-prefixSpan*, and results show that *I-PrefixSpan* outperforms *I-Apriori* both in computing time and scalability. The pseudo-code of the *I-PrefixSpan* algorithm is presented in Section 2.

A few years later, [19] pointed out that most algorithms of the literature use time information only as a time constraint or to represent the time-interval between successive items [9]. The novelty of this work is that not only the delay between successive items is taken into account, but also between distant items. The "multi time-interval (MI) sequential pattern" models the time-intervals between all pairs of items within a pattern. Two algorithms have been proposed,

Algorithm 2 I-PrefixSpan (α, l, D)

```

1: Inputs:
2:  $\alpha = \langle E_1\tau_1 \dots \tau_{l-1}E_l \rangle$ : a temporal pattern.
3:  $l$ : the length of  $\alpha$ .
4:  $D$ : a sequential database, or a projected database.
5: Outputs:
6:  $TP$ : the set of all frequent temporal patterns.
7: Method:
8: Scan  $D$  to find each frequent pair  $(\tau_i, E_{i+1})$ , where  $\tau_i \in TI$  is the gap interval between items  $E_{i-1}$  and  $E_{i+1}$ .
9: for all  $(\tau_i, E_{i+1})$  do
10:   add  $\alpha' = \langle E_1 \dots \tau_{i-1}E_i\tau_i E_{i+1} \rangle$  to  $TP$ , as a new temporal pattern.
11: end for
12: for all  $\alpha'$  do
13:   create the  $\alpha'$ -projected database  $D|_{\alpha'}$ 
14:   call I-PrefixSpan( $\alpha', l+1, D|_{\alpha'}$ )
15: end for

```

MI-Apriori and *MI-prefixSpan*, that are highly similar to the *I-PrefixSpan* and *I-Apriori* algorithms.

Discovering time-interval patterns has attracted considerable efforts, due to its widespread applications. However, several challenges remain, such as the definition of the adequate set of intervals (whether manual or automatic), including the problem of the granularity of the intervals.

2.4 Temporal Granularities

As soon as intervals are introduced, an issue arises: how to choose these intervals?

[3] proposes to manage different temporal granularities. An algorithm composed of Timed Automata with Granularities (TAGs), associated with heuristics is proposed. TAGs test whether a candidate time pattern appears frequently in a time sequence. The heuristic allows to reduce the number of candidates. [29] focuses on mining periodic patterns, where interesting periods cannot be defined in advance. Two temporal granules are proposed: a fine-grained granule for hourly periods and a coarse-grained granule for daily periods. The time distribution of different time granularities is then estimated by using a combination of Gaussian distribution.

2.5 Temporal Data Mining in EDM

To the best of our knowledge, little use has been made of Temporal Pattern Mining in the EDM field. [23] takes time into account by evaluating the rate at which students change the learning resources of interest. They progressively improve "when" resources have to be recommended to the student. In a learning context, where students can choose both which and when courses and exams to take, the research work presented in [4] uses time information that corresponds either to the "semester in which the exam was taken" or to the "delay with which it was taken". Using this time information, they then study the course and exam schedule that the students take and understand better students' behaviors. Using clustering and comparison, they are then able to suggest improvements to the scheduling of courses and exams of students.

3. DEFINITIONS AND METHODOLOGY

The previous literature review highlights that time-intervals are mainly adopted to model temporal patterns. The algorithm proposed in [6], *I-PrefixSpan*, has the main advantage to consider intervals as a core element of the patterns and the mining process. Intervals are considered as a constraint about the patterns, not as supplementary information about the patterns. It is the main reason why we choose to adopt this algorithm in our work.

We start by introducing definitions that will be used in the following methodology and in the experiments.

3.1 Definitions

Let $p = \langle E_1 \tau_1 E_2 \dots \tau_{n-1} E_n \rangle$ and $p' = \langle E'_1 \tau'_1 E'_2 \dots \tau'_{m-1} E'_m \rangle$ be two *ti*-patterns and $s = \langle E''_1 E''_2 \dots E''_l \rangle$ be a *s*-pattern, with n (resp. m and l), the length of the pattern p (resp. p' and s). Given these patterns, we put the following definitions. Recall that *TP* is the set of frequent *ti*-patterns and *SP* the set of *s*-patterns.

DEFINITION 3.1. *ti*-form of an *s*-pattern

p is a *ti*-form of s , denoted by $isform(s, p)$ if and only if: $(n = m) \wedge (E_i = E'_i), \forall i \in [1 : n]$.
ti-form(s) is the set of *ti*-forms, in *TP*, of s .

DEFINITION 3.2. *s*-form of a *ti*-pattern

s is a *s*-form of p if and only if: $(n = s) \wedge (E_i = E'_i), \forall i \in [1 : n]$. *s*-form(p) is the (unique) frequent *s*-form, in *SP*, of p .
s-form(P) is the set of frequent *s*-forms (in *SP*) of the set of *ti*-patterns $p \in P$.

DEFINITION 3.3. *s*-equivalence of *ti*-patterns

p and p' are *s*-equivalent, denoted $s\text{-eq}(p, p')$ if and only if: $(n = m) \wedge (E_i = E'_i), \forall i \in [1 : n]$.
In other words, $s\text{-form}(p) = s\text{-form}(p')$.

DEFINITION 3.4. Prefix of a *ti*-pattern

p' is a prefix of p if and only if:
 $(m < n) \wedge (E_i = E'_i) \wedge (\tau_i = \tau'_i), \forall i \in [1 : m]$.

DEFINITION 3.5. Extension of a pattern

p' is an extension of p if p is a prefix of p' . We note $ext(p)$ the set of extensions of p that belong to *TP*.
A similar definition can be put for *s*-patterns.

DEFINITION 3.6. Extended part of a pattern

Let p' be an extension of p . The extended part of p , with respect to p' , is the pattern p'' , where $concat(p, p'') = p'$. Thus, $p'' = \langle E_n \tau'_n E'_{n+1} \dots \tau'_{m-1} E'_m \rangle$.

We note $extPart(p)$ the set of extended parts of p , i.e. the set of patterns that, when concatenated with p , result in a pattern that belongs to *TP*.

A similar definition can be given for *s*-patterns.

Example: Let $p = \langle e_1 I_1 e_0 \rangle$, and $p' = \langle e_1 I_1 e_0 I_2 e_1 \rangle$ be two *ti*-patterns. $p'' = \langle e_0 I_2 e_1 \rangle$ is an extended part of p .

DEFINITION 3.7. Pseudo-equivalence of *ti*-patterns

p and p' are said to be pseudo-equivalent, if and only if: $s\text{-eq}(p, p') \wedge (\tau_n \neq \tau'_n) \wedge (\tau_i = \tau'_i), \forall i \in [1 : n - 1]$, i.e. they differ only in their last time-interval.

3.2 Methodology

To support our hypothesis and identify the actual value of a *ti*-pattern model, we define a methodology. More precisely, this methodology aims at identifying if there actually are temporal regularities between students' activities, if managing temporal activities allows to have a better view of students' future activities, and concretely what type of activities are mined. Recall that mining *ti*-patterns is quite new in educational data mining.

We intend to mine *ti*-patterns in a temporal database D , which is a database made up of temporal sequences. A temporal sequence is an ordered list of events (concretely a list of resources students interacted with) and their associated timestamp. Each temporal sequence represents one student's temporal activities and each student is represented by a unique (and long) sequence.

Our methodology relies on four steps, described hereafter.

3.2.1 Determining the set of time-intervals

Recall that although timestamps are discrete values, their precision is so high that relying on time-point (or gap) patterns will probably only lead to infrequent patterns. For example, two sequences that only differ by one second: $\langle (0, E_1) (3, E_2) \rangle$ and $\langle (0, E_1) (4, E_2) \rangle$ will correspond to two different patterns. Grouping gaps to form *ti*-patterns, will increase the support of patterns. In addition, if the intervals are appropriate, the loss of precision about temporal activities will be limited.

So, before assessing the relevance of mining *ti*-patterns, we have to choose the adequate set of time-intervals. Indeed, this set influences the information conveyed.

Let $TI = \{I_0, I_1, \dots, I_t\}$ be a set of time-intervals, where $I_j = [gapmin_j; gapmax_j[$ is an interval that contains all gap values between $gapmin_j$ and $gapmax_j$. Notice that the set of intervals should represent a continuum of gap values from $gapmin_0$ to $gapmax_t$.

We propose to evaluate the quality of a set of intervals TI with 3 criteria:

The fitting ratio. It is the ratio between the number of non-empty intervals and the total number of intervals. A non-empty interval is an interval that is part of frequent patterns. The higher the ratio, the better the set of intervals, as the number of "useless" intervals is low.

The number of intervals. On the one hand, the more intervals, the higher the potential of the model. Notice that when $TI = \{I_0\} = [0 : +\infty[$, it comes down to *PrefixSpan*. On the other hand, using too many intervals increases the complexity of the model. In addition, as there are many intervals, the *ti*-patterns discovered will probably be infrequent. Thus, a good set is a set that has an in-between number of patterns.

The horizon. It is represented by TI , the upper bound of the last interval (the maximal time value of the set of intervals). The larger the horizon, the more complete the model, as it is able to represent long-term recurrences.

From our point of view, the best set of intervals is the one that maximizes the fitting ratio while having a large horizon, with a limited number of intervals.

3.2.2 Comparing sets of s -patterns and ti -patterns

After having fixed the set of intervals, the set TP of ti -patterns can be mined. In this second step, we aim at comparing the set TP with the set SP set of s -patterns, and propose some measures to perform this comparison.

First of all, we propose to study the number of patterns, and their average length, to get a coarse-grained view of the set of patterns. Of course, this measure cannot be used alone, as the goal is definitely not to mine the highest number of patterns.

Second, we study the correspondence between both sets of patterns. Let us start by noticing that the number of ti -patterns cannot be deduced (not even approximately) from the number of s -patterns. A brief explanation follows.

Let s be a frequent s -pattern and $ti\text{-cand}(s) = \{ts_1, ts_2, \dots, ts_k\}$ the set of the candidate- ti -forms of s . Note that ts_i may be infrequent. Two cases arise:

- $|ti\text{-cand}(s)| = 1$. This case occurs when all the occurrences of s have the same candidate ti -form ts . Here, $supp(ts) = supp(s)$, thus ts is also frequent. The corresponding set of s -patterns is noted S^1 .
- $|ti\text{-cand}(s)| > 1$. This case occurs when some occurrences of s have different candidate ti -forms. As a consequence, $\forall i, (supp(ts_i) < supp(s)) \wedge (\sum_{i=1}^k supp(ts_i) = supp(s))$. Here, come three possibilities:
 - $\nexists ts_i, supp(ts_i) > \delta$: there exists no frequent ti -form of s , thus the number of frequent patterns decreases. The associated set of patterns is noted S^0 .
 - $\exists! ts_i, supp(ts_i) > \delta$, thus: $\forall j | \{(1 \leq j \leq |ti\text{-seq}(s)|) \wedge (j \neq i)\}, supp(ts_j) < \delta$. In this case, there exists a unique frequent ti -form of s , the number of patterns remains stable.
 - $\exists(i, j), (i \neq j) \wedge (supp(ts_i) > \delta) \wedge (supp(ts_j) > \delta)$. In this case, there exist several frequent ti -forms of s , the number of patterns increases. The set of patterns associated with both last cases is noted S^{1+} . Based on this, we first introduce the pattern loss measure, that represents the ratio of s -patterns that have no ti -form in TP ($s \in S^0$).

$$pLoss(SP) = \frac{|SP| - |\bigcup_{p \in TP} s\text{-form}(p)|}{|SP|} \quad (1)$$

To complete the pattern loss measure, we define the support loss measure, which applies for any s -pattern that has at least one frequent ti -pattern ($s \in S^{1+}$). The support loss measure evaluates the proportion of "lost" occurrences of s , i.e. that have no correspondence in TP .

Let s be a s -pattern and $P = \{p_1, p_2, \dots, p_k\}$ be a set of ti -patterns, where $isform(s, p_i), \forall p_i \in P$. The support loss of s is defined in equation (2).

$$sLoss(s) = \frac{supp(s) - supp^*(P)}{supp(s)} \quad (2)$$

where $supp^*(\cdot)$ is the support of a set of patterns, defined in equation (3).

$$supp^*(P) = |\bigcup_{p \in P} Seq_id(p)| \leq \sum_{p \in P} |supp(p)| \quad (3)$$

where $Seq_id(p)$ is the set of sequence ids in D , where p is a subsequence. We can see that the support of P is not

defined as the sum of the supports of the patterns in P . To explain this, let us consider $P = \{p_1, p_2\}$, with p_1 and p_2 two s -equivalent ti -patterns.

By definition, the s -form of p_1 (which is the same as the s -form of p_2) occurs at most once in each sequence of D . Similarly, p_1 and p_2 occur at most once in each sequence, but both can occur in the same sequence. As a consequence, the support of P may be lower than the sum of the supports of p_1 and p_2 .

The support loss defined above applies for a s -pattern. If the support loss has to be evaluated on a set of patterns, the average support loss and the associated standard deviation can be used.

3.2.3 Evaluating the impact of time on the set of possible future activities of students

In the following third and fourth steps, we aim to evaluate the benefit brought by time in patterns (through ti -patterns) about the possible future activities of students. To perform this evaluation, we adopt a two-stage approach.

Let p be a ti -pattern and $extPart(p)$ the set of extended parts of p (see Def. 3.6). From the educational point of view, the set of extended parts of a ti -pattern p represents the ti -activities that students frequently do after p .

In this third step, we aim at discovering if managing time allows to reduce the uncertainty about the future activities of students. We compare the set of extended parts of s -patterns and the set of extended parts of their ti -forms.

To conduct this comparison, we propose to use the well-known entropy measure. The entropy of a pattern p represents the "degree of disorder" of the set of its extended parts. From the educational view, given an activity performed by students, the entropy measures the uncertainty of its following activities. The higher the entropy, the more uncertain the following activities. Relying on the entropy is not new in the educational field [38]. Equation (4) presents the way the entropy of a ti -pattern p is evaluated.

$$Ent(p) = - \sum_{j=1}^m prob(p_j) \log_2(prob(p_j)), \quad (4)$$

with $prob(p_j) = \frac{supp(p_j)}{\sum_{k=1}^m supp(p_k)}$ and p_j is one of the m extended parts of p . The same equation stands for s -patterns. Given a s -pattern s , we thus propose to evaluate the benefit of considering time-intervals in this pattern, by evaluating the entropy loss (see Equation 5). Entropy loss of an s -pattern s considers the entropy of s ($Ent(s)$) and the maximum entropy of its ti -form.

$$eLoss(s) = \frac{Ent(s) - \max_{p \in ti\text{-form}(s)} \{Ent(p)\}}{Ent(s)} \quad (5)$$

Several cases may arise. First, $eLoss = 1$. This represents the best case: each of the ti -forms of s has exactly one extension. This means that when managing time in patterns, the future activities are totally certain.

Second, $eLoss = 0.0$. This case represents one of the worst cases: at least one ti -form of s has the same entropy as s . Here, we cannot say that managing time makes the possible future activity less uncertain.

Last, $eLoss < 0.0$. This case represents the other worst case: all the ti -form of s has an entropy higher than s . In this case, considering time decreases the quality of the model. Notice here that the term *Loss* is a misnomer as it may theoretically be < 0.0 . However, this term has been chosen to be coherent with previous measures.

As a consequence, the higher the entropy loss ratio the more managing time in patterns contributes to better estimate students' future activities.

3.2.4 Evaluating the impact of a specific time-interval on students' future activities

This fourth and last step is dedicated to the evaluation of the impact of a specific time-interval of a ti -pattern on its extended parts. More precisely, we are interested in the impact of the last time-interval of a pattern. We focus on the following situation: given two pseudo-equivalent ti -patterns (cf., 3.7), to what extent do their set of extended parts differ?

This evaluation allows to study to what extent two students, who perform the same temporal activity, except about the time of their last activity, do have identical future activities. In other words, is a temporal difference between two activities an indicator of activities that are beginning to diverge?

To perform this evaluation, we first evaluate the proportion of identical ti -patterns between pairs of sets of extended parts, as defined in Equation (6).

$$idExt(PQ) = \frac{\sum_{(P,Q) \in PQ} \frac{|P \cap Q|}{|P \cup Q|}}{|PQ|} \quad (6)$$

with $PQ = \{(extPart(p), extPart(q)) | psd-eq(p, q)\}$ the pairs of sets of extended parts of all pseudo-equivalent pairs of ti -patterns. The higher this proportion, the lower the impact of the last time-interval.

Second, we rely on the proportion of s -equivalent extended parts. This measure also evaluates the impact of the last time-interval on the set of extended parts, but by considering only their sequential nature. The proportion of s -equivalent extended parts is defined in Equation (7).

$$sidExt(PQ) = \frac{\sum_{(P,Q) \in PQ} \frac{|s\text{-form}(P) \cap s\text{-form}(Q)|}{|s\text{-form}(P) \cup s\text{-form}(Q)|}}{|s\text{-form}(P, Q)|} \quad (7)$$

This proportion represents if students tend to share their following sequential activities, even though they differ in their last time-interval. Here also, the higher this proportion, the lower the impact of the time-interval.

Notice that for reasons of readability, $s\text{-form}(\cdot)$ is used here to represent the sequential form of a set of ti -patterns and a set of pairs of ti -patterns.

4. EXPERIMENTS

We apply the methodology described in the previous section to evaluate to what extent mining ti -patterns increases the knowledge about students' activities. We first present the dataset on which the experiments are conducted, then use the 4 steps of the methodology and draw conclusions for each of them. Finally, some mined ti -patterns are displayed.

4.1 Dataset overview and implementation

We collected data from 113 first-year university students, enrolled in a Mathematics and Computer Science Bachelor program and who interact with learning resources on their LMS. We focus on one specific course: algorithms and programming from the Fall semester in 2018. This course is a core course of this program. Diverse online materials are available: slides, exercises for lab sessions, tests, etc.

Most of the students own a personal computer, so they can access the course both during teaching hours (lectures or lab sessions) and after official teaching hours.

The set of events E is made of 35 learning resources, that students can consult. About 50% of these resources are studied during the teaching hours (lectures or lab). The dataset is made up of about 6,300 actions and each student sees on average 56 resources. The dataset spans almost one year, as it includes actions performed not only during the teaching period but also during revisions for the final examination and actions conducted for the retake examination (for the subset of students who failed the final examination).

In the experiments conducted, we use a relative minimum support $\delta = 0.1$. Two algorithms are studied: *PrefixSpan*, to mine sequential patterns and *I-PrefixSpan*, to mine ti -patterns. The source code used for *I-PrefixSpan* algorithm is the one available in [12] (we have slightly adapted the code to our needs). The source code used for the classical *PrefixSpan* algorithm is the one proposed by Gao [14].

4.2 Determining the set of time-intervals

We propose to study two types of intervals: Linear intervals, where each interval has an equal duration, and granular intervals, where the duration of intervals grows with the gap value.

Table 1 presents various sets of intervals studied. For each of them, the number of intervals, the maximal horizon, the fitting of the set, the frequency of each frequent interval, as well as the number of frequent patterns, are displayed. To avoid an artificially high fitting value, we consider that an interval is frequent if its frequency is no less than 10. The frequency of an interval is evaluated as the number of times the interval is used in the frequent patterns.

Before going into the details of the analysis of the set of intervals, we would like to mention that the sets do not all have the same number of intervals, so these values in Table 1 are not directly comparable. In addition, two contiguous granular intervals represent a totally different duration (for example up to 1 hour and up to 1 day), the frequencies are therefore not comparable. Last, notice that the total number of patterns in one set of intervals cannot be explained by the number of patterns of another set. Let us for example consider two sets of intervals and their associated number of patterns. Suppose that the first interval has an average duration twice longer than the second one. A pattern that is frequent in the first set may correspond to either two frequent patterns in the second set, or only one frequent pattern, or no frequent pattern at all (see section 3.2.2).

Let us first consider the three sets of linear intervals. For the two first sets (30 min and 1 hour), the fitting measure is quite low: 8%, which means that the vast majority of in-

Type	Duration	Number of intervals	Horizon	Fitting	Used intervals & associated frequency	# patterns
Linear	30 min	25	12h.	8%	$I_0 : 350,000 ; I_{24} : 1,770,000$	550,000
Linear	1 hour	25	12h.	8%	$I_0 : 549,380 ; I_{24} : 1,843,660$	356,811
Linear	1 day	25	24d.	72%	$I_0 : 90,976 ; I_1 : 42 ; I_2 : 25 ;$ $I_3 : 54 ; I_4 : 119 ; I_5 : 36 ;$ $I_6 : 239 ; I_7 : 189 ; I_8 : 17 ;$ $I_{11} : 14 ; I_{12} : 44 ; I_{13} : 80 ;$ $I_{14} : 17 ; I_{17} : 14 ; I_{18} : 13 ;$ $I_{20} : 36 ; I_{21} : 45 ; I_{24} : 33,298$	37,764
Granular	expon.	16 $I_0 = [0 \text{ sec.} ; 10\text{mn.}]$	8mt.	56%	$I_0 : 17,739 ; I_1 : 79 ; I_8 : 82 ; I_9 : 269 ;$ $I_{10} : 4,278 ; I_{11} : 5,126 ; I_{12} : 6,403 ;$ $I_{13} : 3,693 ; I_{14} : 1,159$	15,754
Granular	human	6 $I_0 = [0 \text{ sec.} ; 1 \text{ mn.}]$	1y.	100%	$I_0(\text{sec}) : 7,706 ; I_1(\text{min}) : 10,551 ;$ $I_2(\text{hour}) : 1,615 ; I_3(\text{day}) : 30,925 ;$ $I_4(\text{week}) : 68,614 ; I_5(\text{month}) : 22,479$	51,025

Table 1: Fitting, examples of intervals and number of patterns for several sets of intervals

tervals are not found in frequent patterns. For example, in "30 min", only the first interval (between 0 and 30 minutes) and the last interval (more than 12 hours) are not empty. We can conclude that both sets of intervals are not good candidates. Caution must be exercised in interpreting this result. It might mean that students do not regularly switch from one resource to another, with a time gap between 30 minutes and 12 hours. It can also mean that the 30 min. time-interval is not relevant. Despite the lack of relevance of these intervals, the number of patterns discovered is important. As only two interval patterns are used, we can consider that *I-PrefixSpan* behaves here almost as PrefixSpan.

The fitting value of the "1 day" set is quite larger: 72%, which means that most of the 25 intervals are frequent. However, the total number of frequent patterns in this set is highly decreased, compared to the "30 min" and "1h" sets (by about 10 times). In addition, many interval frequencies are not so high, some of them being close to the minimal threshold, except the first and last one. This tends to mean that many intervals are not that representative of the data. Moreover, although the number of intervals is quite large (25), the maximal horizon represented by this set remains limited (all together, except the last one, represent a horizon of smaller than a month). Recall that the dataset spans almost one year. Obviously, the horizon can be extended, but it will be at the cost of an even larger number of intervals, as well as an increase in the space and computation time. These results tend to suggest that the set of intervals should contain small intervals for close events (such as suggested by the frequency of I_0 in the 30 min set), and larger intervals for furthest gaps (such as suggested by the frequency of I_{24} in the 1 day set). Thus, a granular set of intervals should better fit the dataset.

We propose to study now two sets of granular intervals. In the first set, the duration of intervals grows exponentially: the duration of an interval is twice larger than the duration of the preceding interval. The fitting of this set is greater than for the two first ones, but smaller than the third one. Nevertheless, the horizon is larger than for all the previous ones (about 4 months), and the number of intervals is decreased. The empty intervals (from I_2 to I_7) tend to represent a gap between 20 min and 10 hours 40 min.

The second set of granular intervals is referred to as "human", the intervals are designed to represent the human natural time: minute, hour, day, week, etc. This set of intervals has a maximal fitting (100%). At the opposite of the "1 day" intervals, that has the highest fitting value till then, the frequency of each interval is quite large (greater than 1,600) and the number of intervals is reduced (only 6 intervals). Besides, the total number of patterns is larger than both the "1 day" and the "exponential" sets.

All these elements contribute to consider the "human" set as the best set of intervals. In this set, time is represented by the {minute, hour, day, week, month, year} intervals. This set has a maximal fitting (100%), covers a large horizon (till a year, which corresponds to the span of the dataset), with a limited number of intervals (6 intervals) and provides a quite large number of frequent temporal patterns. Therefore, in the following experiments, this set of intervals will be used.

Given these elements, we would like to highlight that this set of intervals intrinsically represents the classical rhythm of courses, for example one lecture (or one lab session) is planned each week. The human set of intervals thus allows to mine patterns that represent natural students temporal activities: some students tend to work immediately following a lab session (or a lecture) represented by I_0 or I_1 ; other students wait for some hours in the same 24h, and others work during the week, or even the week after (before the next session) represented by I_4 . It is typically the type of information that we expect to get when we aim at modeling students' activities.

4.3 Comparing s-patterns and ti-patterns

This second experiment aims at comparing sets of *s*-patterns and *ti*-patterns. Table 2 presents both sets of patterns, associated with measures introduced in the methodology. Let us first focus on the number of frequent patterns (line 1). The total number of frequent *ti*-patterns is dramatically smaller than the number of frequent *s*-patterns. The pattern loss is larger than 0.99. This means that the great majority of *s*-patterns has no frequent *ti*-forms, probably due to the spread of occurrences of *s*-patterns over numerous *ti*-patterns. These findings are in line with [22]. In addition,

$\delta = 0.1 (= 11)$	<i>PrefixSpan</i>	<i>I-PrefixSpan</i>
Number of patterns	$ SP = 12, 826, 760$	$ TP = 51, 025 - pLoss(SP) = 0.998$
Average	8	3.8
Max. length	17	8
Example of pattern s, $frequent(s)$ $\wedge \{ \#p \mid (isform(s, p) \wedge frequent(p)) \}$	$s = \langle e_{31}, e_{29} \rangle$ $supp(s) = 26$	$ts_1 = \langle e_{31} I_0 e_{29} \rangle, supp(ts_1) = 1$ $ts_2 = \langle e_{31} I_1 e_{29} \rangle, supp(ts_2) = 10$ $ts_3 = \langle e_{31} I_2 e_{29} \rangle, supp(ts_3) = 6$ $ts_4 = \langle e_{31} I_3 e_{29} \rangle, supp(ts_4) = 3$ $ts_5 = \langle e_{31} I_4 e_{29} \rangle, supp(ts_5) = 4$ $ts_6 = \langle e_{31} I_5 e_{29} \rangle, supp(ts_6) = 7$
Example of pattern s, $frequent(s)$ $\wedge \{ \exists (p_i, p_j) \mid (frequent(p_i) \wedge frequent(p_j)) \}$	$s = \langle e_{22}, e_{33} \rangle$ $supp(s) = 53$	$p_1 = \langle e_{22} I_0 e_{33} \rangle, supp(p_1) = 25$ $p_2 = \langle e_{22} I_1 e_{33} \rangle, supp(p_2) = 22$
Support loss		$sLoss(SP^{1+}) = 0.33$; $std(sLoss(SP^{1+})) = 0.10$

Table 2: Comparison of sets of patterns mined with *PrefixSpan* and *I-PrefixSpan*

we can see in Line 2 that the average length of *ti*-patterns is about twice smaller than the length of *s*-patterns, the same for their maximal length. A first conclusion that can be drawn here is that most of frequent sequential patterns have no recurrences in their time-intervals. This means that students tend to have numerous recurrent sequential activities, and quite less recurrent time-interval activities. However, even though the average length of patterns is divided by 2, *ti*-patterns have a significant length, which means that they do represent a meaningful students' activities. Moreover, a tens of thousands *s*-patterns (about 34,000) have one or more frequent *ti*-forms (about 51,000). This means that for these sequential activities, there are actually temporal regularities. These activities will be studied in more detail in the following section.

Lines 4 and 5 in Table 2 illustrate some examples of *s*-patterns and their candidate or frequent *ti*-forms. Line 4 presents one of the 99.8% *s*-patterns that has no *ti*-form (thus, from SP^0). This pattern ($s = \langle e_{31}, e_{29} \rangle$) has 6 candidate *ti*-forms, but none of them is frequent. We can conclude that no obvious time-interval regularity is observed for this activity. Thus, this activity does not seem to be guided by temporal constraints. We can also observe here that the sum of the support of the *ti*-patterns is greater than the support of their *s*-form *s*. This was mentioned in section 3.2.2.

In the remaining sequential patterns (SP^{1+}) made up of about 34,000 *s*-patterns, 65% of the *s*-patterns have exactly 1 frequent *ti*-form and 91% have 1 or 2 frequent *ti*-forms. The highest number of frequent *ti*-forms of an *s*-pattern is 9, which is quite high. Let us now consider line 5 in Table 2, that presents a *s*-pattern that has several frequent *ti*-forms. This *s*-pattern ($s = \langle e_{22}, e_{33} \rangle$) has a support equal to 53 and two frequent *ti*-forms. Such a pattern occurs with two temporal recurrences, and most of its occurrences have a time gap between 1 minute and 1 day. Such patterns are highly interesting and will also be further studied.

Based on these findings, it is legitimate to ask whether a *ti*-pattern-based model can replace a *s*-pattern-based model. Line 1 gives first indications. Many sequential patterns "disappear" with such a model (more than 99% of sequential patterns have no frequent *ti*-pattern). If the objective is to replace traditional *s*-patterns by *ti*-patterns, a problem of coverage of the model arises. However, if the goal is to iden-

tify which activities (sequential) have temporal regularities, *ti*-patterns are of the highest interest.

Let us now focus on the support loss associated with the complete set SP^{1+} of *s*-patterns that have at least one frequent *ti*-form. $sLoss(SP^{1+}) = 0.33$, with a standard deviation equal to 0.1. This means that on average 1/3 of the occurrences of an *s*-pattern "disappear", i.e. they do not belong to any frequent *ti*-form. We can conclude that among patterns with identified temporal regularities, 33% of the occurrences do not follow this regularity, which may be high.

4.4 Evaluating the impact of time on the set of possible future activities of students

Following our methodology, we evaluate now if *ti*-patterns carry more information than *s*-patterns about future activities of students. As a preliminary remark, we would like to mention that $\#s, eLoss(s) \leq 0$. We mentioned previously that this case would occur rarely, in practice here it does not occur.

In the set SP^{1+} , 71% of the patterns have at least one extension in SP (see Def. 3.5). Let us first consider the 66% of these patterns that have a unique extension. By definition for these patterns, $Ent(s) = 0$ and $Ent(p) \geq 0, \forall p \in ti\text{-form}(s)$. The first Line of Table 3 is an example of such a case. The *s*-pattern $\langle e_{24} e_{27} e_{14} \rangle$ has only one extended part, so its entropy equals zero. It has three *ti*-forms, but only one has an extended part. So, all these *ti*-forms have an entropy equals to zero.

In this case, even if the entropy loss is null, the information about the future activities of students is increased, as only one *ti*-pattern has a frequent extended part.

Let us now consider the 34% remaining patterns, which have more than one extension in SP . The average entropy is 0.84 with a maximal entropy of 7.71. When focusing on the set of their *ti*-forms, the average entropy is 0.35 and the maximal entropy is 6.22. To make entropies as comparable as possible, the average entropy for *s*-patterns has been evaluated only on the set of *s*-patterns that have at least one *ti*-form. We can first notice that entropy of *s*-patterns is globally higher than the one of *ti*-patterns (for both maximal and average values). More precisely, the average entropy of *s*-patterns is 2.4 times bigger than the one of *ti*-patterns. We can thus draw a first global conclusion: managing time in

s-pattern	Examples of extPart(s)	Ent(s)	Examples of $p \in ti\text{-form}(s)$	nbExt(p)	Examples of extPart(p)	max-Ent(p)	mean-Ent(p)
$\langle e_{24} e_{27} e_{14} \rangle$	$\langle e_{12} \rangle$	0.0	$\langle e_{24} I_2 e_{27} I_3 e_{14} \rangle$ $\langle e_{24} I_3 e_{27} I_3 e_{14} \rangle$	0 1	$\langle e_{14} I_3 e_{12} \rangle^{(*)}$	0.0	0.0
$\langle e_1 e_{10} e_{12} \rangle$	$\langle e_{\{3,13\}} \rangle$ $\langle e_{19} e_{\{3,22\}} \rangle$ $\langle e_{12} e_{\{12,19\}} \rangle$	5.49	$\langle e_1 I_5 e_{10} I_1 e_{12} \rangle$ $\langle e_1 I_5 e_{10} I_0 e_{12} \rangle$ $\langle e_1 I_5 e_{10} I_2 e_{12} \rangle$ $\langle e_1 I_2 e_{10} I_2 e_{12} \rangle$	0 1 13 24	$\langle e_{12} I_4 e_3 \rangle$ $\langle e_{12} I_5 e_{\{19,13,12\}} \rangle$ $\langle e_{12} I_5 e_{\{19,13,15\}} \rangle$	4.55	0.78

Table 3: Examples s-patterns with ti-forms, extended parts and entropy values. (*) The corresponding extension pattern is $p'' = \langle e_{24} I_3 e_{27} I_3 e_{14} I_3 e_{12} \rangle$.

patterns allows to decrease the uncertainty of students' future activities.

We will now compare the entropy of each s-pattern, with the entropy of its ti-forms (through $eLoss$). In 68% of the cases, the entropy loss between the s-patterns and their ti-forms is higher than 0. This means that when considering a temporal student activity, in 2 cases out of 3, the future activity of this student is less uncertain than when managing his/her sequential activity. These 68% are divided into 51% with a loss equal to 1, which means that future activities become certain. 17% of the cases have a loss between 0 and 1. The average entropy loss on all s-patterns is quite high: $eLoss = 0.4$. Roughly speaking, the future activities of students are on average 40% less uncertain when managing time in patterns, which is highly promising.

Thanks to these experiments, we confirm that managing time-interval patterns allows, in most cases, to have a better view of the following activities of students. In addition, for a significant number of activities, future activities are now totally certain.

Let us now focus on an example presented in the second Line of Table 3. The s-pattern $s = \langle e_1 e_{10} e_{12} \rangle$ has many extensions in SP and many ti-forms, among which many of them have extensions. Notice that although the entropy loss is low (the maximal entropy of the ti-forms is 4.55), on average it is significantly lower (0.78). In this specific case, $eLoss$ measure is not that representative of the difference in entropy, the entropy decrease is probably higher than the $eLoss$ value.

4.5 Evaluating the impact of a specific time-interval on students' future activities

The experiments conducted here fall within the scope of the last step of our methodology. They aim at evaluating to what extent two students who perform a similar activity (both in terms of resources and time-interval) and who only differ in their last time-interval, have the same future activities. In the experiments conducted, we will only focus on patterns made up of at least 3 events (and 2 time-intervals) to ensure that the patterns can be considered as activities.

In the set PQ composed of $|PQ| = 9,510$ of pseudo-equivalent pairs of patterns (cf., Definition 3.7), 25% of the extended parts of a pattern of any pair are also part of the extended parts of the other pattern (sequentially and temporally identical). 11% additional pairs have sequentially identical extended parts. This highlights that even when two ti-patterns

differ in their last time-interval only, this small difference leads to a significant difference in their sets of extended parts. In terms of students' activities, this means that when two students make exactly the same activity, except on the last time-interval, their following activities mainly differ: not only in terms of temporal activities but also in terms of their sequential activities. We can conclude that the last time-interval highly influences students' future activities and that it may be viewed as an indicator of activities that are beginning to diverge.

Experiments conducted in both previous sections confirm that ti-patterns contribute to the increase of the information about students' future activities whereby the uncertainty of this future is reduced. As a consequence, we can say that time is an important information in students' activities.

4.6 Interpretation of ti-patterns

In this section, we present examples of frequent ti-patterns, in an understandable format to better analyze and understand students' activities.

The events ids in patterns are replaced by their type and an id. Lec_n will refer to the slides associated with the n^{th} lecture; $Glos_n$ will be the n^{th} glossary resource; Stx_n a syntax resource; Sum_n will be a summary resource; Lab_n a resource that contains exercises that are studied during lab sessions (exercise sheets); FA_n are facultative additional exercises; finally Ad is the advise resource. The time-intervals are noted $(I_s, I_{mn}, I_h, I_d, I_w, I_{mt})$, which refer to seconds, minutes, hours, days, weeks and months.

Given that the longer an activity, the more information it contains, we will preferably focus on the longest ti-patterns.

Activities made up of temporally close events

Let us start by studying activities that contain only the "seconds" time-interval (i.e. events with a maximal gap of 1 minute). This will allow us to have a better view of the type of activities that are performed on the spot. First, the corresponding activities tend to be made up of specific types of events: they are a mix of glossary, syntax, advertisement and lab resources. Second, the maximum length here is 7, which means that there are actually long recurrent "quick" activities made by students. Third, when analyzing the activities, we can remark that they all have a similar skeleton: students generally start by looking at the following resources (in any order): $\{Sum_5, Stx_3, Glos_3\}$, then study one or more Lab exercises and finally consult an advice page. Let us for example present a ti-pattern of length 7:

$\langle Sum_3 I_s Stx_3 I_s Glos_3 I_s Lab_1 I_s Lab_2 I_s Lab_3 I_s Ad \rangle$

Such patterns can be interpreted as follows: they represent

typical activities performed when preparing an exam. Not only several *Lab* sheets are studied, but also before these resources, students have a quick look at the syntax, glossary, and summary of the lectures. They finally consult the advice page. In such patterns, students interact with resources within a short time, including with the lab sheets.

Activities made up of I_h time-intervals

Let us now focus on patterns that use the "hours" time-interval, where patterns are made up of events with a gap value between 1 hour and 1 day. Here again, we identify a skeleton shared by most of the *ti*-patterns:

$\langle Lab_{\{1,3\}} I_h Lab_{\{2,3\}} I_h Lab_{\{2,3,4\}} \rangle$, where $Lab_{\{1,3\}}$ means either Lab_1 or Lab_3 .

These patterns highlight that some students tend to work sequentially on several exercise sheets. The gap being between 1 hour and 1 day, tends to mean that students dig deep into their works: they spend some hours to perform each exercise sheet.

Other intervals

We have performed a similar study on other time-intervals. For each of them, we also identify skeletons shared by almost all the patterns.

An interesting conclusion that can be drawn from these findings is that for any given time-interval, typical long activities are made by students, that do all have the same skeleton. More importantly, when comparing skeletons between time-intervals, they are totally different. We can thus conclude that the type of activity performed is strongly linked with the "rhythm" of the activity. Here, "rhythm" means a time-interval granularity shared by all gap between all events of the activity.

Last, when studying the timestamps associated with each occurrence of the activities presented above, there is no specific period associated: they are performed at any moment in the semester. For example, when considering the first example given, that is mainly related to the 3^{rd} lecture, we found similar patterns for the 1^{st} , 2^{nd} , etc. lecture resources.

5. DISCUSSION

While traditional studies emphasize that students have typical sequential learning behaviors (identified by frequent sequential patterns), this study further emphasizes that for specific activities students work with temporal regularities. Based on the experiments conducted in the previous sections, we initiate a discussion.

The results have highlighted that among the sets of intervals tested (linear and granular), the one that represents the human natural time is the most relevant one, at least for the dataset used in the experiments (see section 4.2). In addition to outperforming other sets of intervals according to predefined measures, this set conforms to the scope of application: the duration of most of the lectures or lab sessions is about one hour, two successive lectures tend to occur each week, etc. So, the interpretation of the discovered patterns is enhanced. Of course, many other sets of intervals remain untested and may be more adequate. Besides, an automatic approach that learns the optimal set of intervals could be tested, as in [24]. However, this would be at a significant additional computational cost, without any guarantee of applicative interpretability of these intervals.

As expected, a high number of sequential patterns have no frequent *ti*-form. In the experiments conducted, we have even highlighted that most of the sequential activities have no temporal regularities. This results in a high number of "lost" patterns, which can be problematic, in case we are interested in both frequent *ti*-patterns and *s*-patterns. A solution could manage both types of patterns: sequential students' activities mixed with temporal students' activities. This solution would not only maintain the coverage of the model, thanks to sequential patterns but also manage time, thanks to temporal patterns, when suitable. Here is an example of such a pattern: $\langle E_2 E_{27} I_1 E_{13} \rangle$. This pattern means that many students consult E_2 then E_{27} (whatever is the time-interval), then between 1 minute and 1 hour later they do consult E_{13} .

Focusing on *s*-patterns and their various frequent *ti*-forms can help to highlight different learning approaches adopted by students. For example, an activity done with a gap lower than 1 minute between its events may represent the fact that the associated students are used to first download all the resources and then work offline. The same activity with a time gap between 1 minute and 1 hour may reflect that students do work online, they do not access a resource before finishing the previous one. So, in addition to highlighting the diversity of activities of students, *ti*-patterns are also a way to identify students' learning practices. One can foresee that these patterns could be used as input information for many works such as those that focus on students' engagement.

6. CONCLUSION AND FUTURE WORKS

The study presented in this paper highlights the relevance of using time information when mining patterns of students' activities. A time-interval pattern mining approach, through the *I-PrefixSpan* state-of-the-art algorithm, has been adopted to conduct this study.

The experiments conducted have pointed out that the nature of the set of intervals used highly impacts the representativity of the model and that the set of intervals that represents the human natural time is adequate. We also found that most of the sequential students' activities do not correspond to any time-interval activity. However, for other cases, managing this time-interval provides a better view of the future possible students' activities, thanks to temporal indicators. Moreover, results show that a single time-interval difference between two events of two patterns sequentially equivalent results in significantly different subsequent activities.

We thus confirm our hypothesis: temporal information is highly promising for a more precise modeling of students' activities. One additional experiment has illustrated some frequent students' activities both temporal and sequential. It has put forward that, by looking at some specific time-intervals, we can understand what activities students often perform instantly or throughout a longer period.

The work we have conducted provides a first step towards longer-term research. One of our future goals is to provide students with recommendations of educational resources. By relying on *ti*-patterns, we are confident that not only the accuracy of the recommendations provided to students will be increased but also that these patterns will give indications about the right time to propose recommendations to students.

7. REFERENCES

- [1] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. 11th Int. Conf. on Data Engineering*, volume 95, pages 3–14, 1995.
- [2] N. Anjum and S. Badugu. A study of different techniques in educational data mining. In *ICETE*, pages 562–571. Springer, 2020.
- [3] C. Bettini, X. S. Wang, S. Jajodia, and J. Lin. Discovering frequent event patterns with multiple granularities in time sequences. *Trans. Knowledge Data Engineering*, 10(2):222–237, 1998.
- [4] R. Campagni, D. Merlini, R. Sprugnoli, and M. C. Verri. Data mining models for student careers. *Expert Systems with Applications*, 42(13):5508–5521, 2015.
- [5] R. Cerezo, M. Sanchez-Santillan, J. Nunez, and M. P. Paule. Different patterns of students interaction with moodle and their relationship with achievement. *Computer Science*, 2015.
- [6] Y. Chen, M. Chiang, and M. Ko. Discovering time-interval sequential patterns in sequence databases. *Expert Systems with Applications*, 25(3):343–354, 2003.
- [7] Y. Chen, P.-H. Willemin, and J.-M. Labat. Discovering prerequisite structure of skills through probabilistic association rules mining. *International Educational Data Mining Society*, 2015.
- [8] Y.-C. Chen, W.-C. Peng, and S.-Y. Lee. Mining temporal patterns in time interval-based data. *TKDE*, 27(12):3318–3331, 2015.
- [9] Y.-L. Chen, M.-C. Chiang, and M.-T. Ko. Discovering time-interval sequential patterns in sequence databases. *Expert Systems with Applications*, 25(3):343–354, 2003.
- [10] S. Doroudi, K. Holstein, V. Aleven, and E. Brunskill. Sequence matters, but how exactly? A method for evaluating activity sequences from data. In *Proc. 9th IEDMS, 2016*, pages 70–77, 2016.
- [11] S. Fatahi, F. Shabanali-Fami, and H. Moradi. An empirical study of using sequential behavior pattern mining approach to predict learning styles. *Education and Information Technologies*, 23(4):1427–1445, 2018.
- [12] P. Fournier-Viger, J. C. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam. The SPMF open-source data mining library version 2. In *Proc. Int. Conf. ECML PKDD, Riva del Garda, Italy*, volume 9853, pages 36–40. Springer, 2016.
- [13] D. Fricker, H. Zhang, and C. Yu. Sequential pattern mining of multimodal data streams in dyadic interactions. In *1st ICDL-EPIROB*, pages 1–6, 2011.
- [14] C. Gao. Prefixspan-py, 2015–2020.
- [15] F. Giannotti, M. Nanni, D. Pedreschi, and F. Pinelli. Mining sequences with temporal annotations. In *Proc ACM SAC, Dijon, France*, pages 593–597, 2006.
- [16] S. Gutierrez-Santos, M. Mavrikis, and G. Magoulas. Sequence detection for adaptive feedback generation in an exploratory environment for mathematical generalisation. In *Int. Conf. on AIMS*, pages 181–190. Springer, 2010.
- [17] T. Guyet and R. Quiniou. Mining temporal patterns with quantitative intervals. In *Proc. Int. Conf. on Data Mining Workshops*, pages 218–227, 2008.
- [18] Y. Hirate and H. Yamana. Generalized sequential pattern mining with item intervals. *JCP*, 1(3):51–60, 2006.
- [19] Y. Hu, T. C. Huang, H. Yang, and Y. Chen. On mining multi-time-interval sequential patterns. *Data Knowledge Engineering*, 68(10):1112–1127, 2009.
- [20] J. Kay, N. Maisonneuve, K. Yacef, and O. Zaïane. Mining patterns of events in students’ teamwork data. In *Proc. Workshop on EDM at the 8th Int. Conf. on Intelligent Tutoring Systems*, pages 45–52, 2006.
- [21] J. S. Kinnebrew, K. M. Loretz, and G. Biswas. A contextualized, differential sequence mining method to derive students’ learning behavior patterns. *J. of Educational Data Mining*, 5(1):190–219, 2013.
- [22] H. Kitakami, T. Kanbara, Y. Mori, S. Kuroki, and Y. Yamazaki. Modified prefixspan method for motif discovery in sequence databases. In *Proc. Pacific Rim International Conference on Artificial Intelligence*, pages 482–491. Springer, 2002.
- [23] C. Krauss, A. Merceron, and S. Arbanowski. The timeliness deviation: A novel approach to evaluate educational recommender systems for closed-courses. In *Proc. 9th Int. Conf. on LAK, Tempe, USA*, pages 195–204, 2019.
- [24] S. Mahajan and A. Reshamwala. An approach to optimize fuzzy time-interval sequential patterns using multi-objective genetic algorithm. In *Technology systems and management*, pages 115–120. Springer, 2011.
- [25] R. Martínez Maldonado, K. Yacef, J. Kay, A. Kharrufa, and A. Al-Qaraghuli. Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In *Proc. 4th Int. Conf. on Educational Data Mining, Eindhoven, The Netherlands*, pages 111–120, 2011.
- [26] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proc. 17th Int. Conf. on Data Engineering*, pages 215–224, 2001.
- [27] D. Perera, J. Kay, I. Koprinska, K. Yacef, and O. R. Zaïane. Clustering and sequential pattern mining of online collaborative learning data. *IEEE Trans. Knowl. Data Eng.*, 21(6):759–772, 2009.
- [28] L. K. Poon, S.-C. Kong, M. Y. Wong, and T. S. Yau. Mining sequential patterns of students’ access on learning management system. In *Int. conf. on data mining and big data*, pages 191–198. Springer, 2017.
- [29] P. Rashidi and D. J. Cook. Keeping the resident in the loop: Adapting the smart home to the user. *IEEE Transactions on systems, man, and cybernetics-part A: systems and humans*, 39(5):949–959, 2009.
- [30] C. Romero and S. Ventura. Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1):135–146, 2007.
- [31] I. Sato, Y. Hirate, and H. Yamana. Text mining using prefixspan constrained by item interval and item attribute. In *Proc. 22nd ICDE, Atlanta, USA*, page 118, 2006.
- [32] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Int. Conf. on Extending Database*

- Technology*, pages 1–17. Springer, 1996.
- [33] S. Sumalatha and R. Subramanyam. Distributed mining of high utility time interval sequential patterns using mapreduce approach. *Expert Systems with Applications*, 141:112967, 2020.
 - [34] J. K. Tarus, Z. Niu, and A. Yousif. A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. *Future Generation Computer Systems*, 72, 2017.
 - [35] A. Vautier, M. Cordier, and R. Quiniou. An inductive database for mining temporal patterns in event sequences. In *International Joint Conference On Artificial Intelligence*, pages 1640–1641, 2005.
 - [36] M. Yoshida, T. Iizuka, H. Shiohara, and M. Ishiguro. Mining sequential patterns including time intervals. In *Data Mining and Knowledge Discovery: Theory, Tools, and Technology II, Orlando, USA*, volume 4057, pages 213–220, 2000.
 - [37] M. J. Zaki. SPADE: an efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.
 - [38] G. Zhu, W. Xing, and V. Popov. Uncovering the sequential patterns in transformative and non-transformative discourse during collaborative inquiry learning. *The Internet and Higher Education*, 2019.

Automatic Subject-based Contextualisation of Programming Assignment Lists

Samuel C. Fonseca
Institute of Computing
Federal University of
Amazonas
Manaus, Brazil
scf@icomp.ufam.edu.br

Filipe Dwan Pereira
Department of Computer
Science
Federal University of Roraima
Boa Vista, Brazil
filipe.dwan@ufr.br

Elaine H. T. Oliveira
Institute of Computing
Federal University of
Amazonas
Manaus, Brazil
elaine@icomp.ufam.edu.br

David B. F. Oliveira
Institute of Computing
Federal University of
Amazonas
Manaus, Brazil
david@icomp.ufam.edu.br

Leandro S. G. Carvalho
Institute of Computing
Federal University of
Amazonas
Manaus, Brazil
galvao@icomp.ufam.edu.br

Alexandra I. Cristea
Department of Computer
Science
Durham University
Durham, United Kingdom
alexandra.i.cristea@durham.ac.uk

ABSTRACT

As programming must be learned by doing, introductory programming course learners need to solve many problems, e.g., on systems such as 'Online Judges'. However, as such courses are often compulsory for non-Computer Science (non-CS) undergraduates, this may cause difficulties to learners that do not have the typical intrinsic motivation for programming as CS students do. In this sense, contextualised assignment lists, with programming problems related to the students' major, could enhance engagement in the learning process. Thus, students would solve programming problems related to their academic context, improving their comprehension of the applicability and importance of programming. Nonetheless, preparing these contextually personalised programming assignments for classes for different courses is really laborious and would increase considerably the instructors'/monitors' workload. Thus, this work aims, for the first time, to the best of our knowledge, to *automatically classify the programming assignments in Online Judges based on students' academic contexts* by proposing a *new context taxonomy*, as well as a *comprehensive pipeline evaluation methodology* of cutting edge competitive Natural Language Processing (NLP). Our comprehensive methodology pipeline allows for comparing state of the art data augmentation, classifiers, beside NLP approaches. The context taxonomy created contains 23 subject matters related to the non-CS majors, representing thus a challenging multi-classification problem. We show how even on this problem, our comprehensive pipeline evaluation methodology allows us to achieve

an accuracy of 95.2%, which makes it possible to automatically create contextually personalised program assignments for non-CS with a minimal error rate (4.8%).

Keywords

non-CS majors, NLP, contextually personalised assignment lists

1. INTRODUCTION

Introductory Programming (often known under the label of 'CS1') classes are now-a-days often compulsory for undergraduate courses that do not have computing as their major [10, 15, 20, 23]. CS1 is delivered to students majoring in, e.g., mechanical engineering, economics, etc. - whom we collectively name here 'non-CS students'. It is common in such cases to find students with difficulty in interpreting assignment texts, due to the lack of affinity with the area of the problem [22]. As a result, many of these students may be discouraged by CS1, as they fail to see the purpose that programming can have in their professional lives [10, 17, 23].

Moreover, programming must be learned by doing and, hence, learners need to solve many problems [11, 17–19, 27]. In this sense, 'Online Judge' systems can influence positively the learning process of non-CS students [12, 18, 20, 25], as systems which allow students to submit programming assignments and provide real-time automatic code correction. As Programming Online Judges (POJ) have large numbers of problems registered in their problem banks [25], in principle, there would be plenty of problems to select from, for both students as well as teachers, allowing for a mass personalisation - where one teacher could cater in parallel for the needs of many students. Nonetheless, the problems available on these systems often are collected or scraped from various environments that do not provide labelling [27], and thus it is laborious to find appropriate problems for non-CS students. This is more so the case, as the number of programming exercises is constantly increasing [25, 27]. Therefore,

Samuel Fonseca, Filipe Dwan Pereira, Elaine H. T. Oliveira, David Fernandes, Leandro Carvalho and Alexandra Cristea "Automatic Subject-based Contextualisation of Programming Assignment Lists" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 81 - 91

the automatisisation of the categorisation of problems based on subject matter is becoming vital, to support instructors who teach computer programming disciplines. To illustrate, undergraduate students of Economics would be more familiar with an *if-then-else* problem using terms such as “interest rates” or “importation of goods” instead of a problem on the “growth of cells”, which may be completely out of their comfort zone. Thus, we raise the following research question:

How can we extract the subject matter from programming problem statements, to automatically match programming assignment lists to non-CS courses?

Our main contributions with this paper are thus:

- Proposing a new, *wholistic methodology pipeline for the POJ contextual labelling problem*, allowing to compare a variety of cutting edge shallow and deep learning models, to experiment with the most recent data augmentation techniques (with or without augmentation), NLP (based on BERT, Word2Vec, Glove), classifiers (based on BERT, Random Forest, SVM, XGBoost, GaussianNB, GradientBoosting, ExtraTree, Sequential DNN, CNN, RNN) and validation.
- Extracting, for the first time, to the best of our knowledge, automatically and precisely, subject matters related to non-CS courses; we do this by using cutting edge NLP techniques on the statements of assignments available in a home-made online judge CodeBench¹ used with fifteen non-CS major programmes.
- Proposing a *subject-based contextualisation taxonomy* to map subject matters to non-CS courses, where CS1 is compulsory.
- We thus are enabling the contextual personalisation of programming assignment lists for non-CS courses.

2. RELATED WORK

There are many studies tackling the challenge of teaching introductory programming to non-CS students, based on a variety of angles. To illustrate, [10] employed collaborative scenarios to enhance teaching and learning programming in non-CS courses, whilst [23] used an approach involving games and media. [15, 24] show that English-like (natural language) syntax can help non-CS students overcome the difficulties in learning programming syntax. Furthermore, a recent study [21] explains that effective motivational educational design can enhance introductory programming students and teacher engagement. Despite these works representing a move towards improving non-CS students engagement, linking text collections to general or domain-specific knowledge is essential [1, 5]. More specifically, [14] argue that students’ experiences of the learning context have important implications for teaching and learning. Nevertheless, none of these aforementioned studies take the context of the problem into account. Especially untouched is the issue of contextualisation of the problem statements, ensuring that problems introduce only the degree of difficulty required to progress

¹<http://codebench.icomp.ufam.edu.br/index.php>

in the programming knowledge and not additional complexity from strange contexts for the current learner (such as a geology context for economy students, etc.).

Online judges (POJ) are increasingly being used to support introductory programming (CS1) classes. Via such environments, teachers can provide problems to be solved and students can submit their code and receive immediate feedback [9, 18, 25]. One of the issues of these systems is that, in general, the problems available are not categorised based on subject matter, topics, context, major, etc. In this sense, there are two recent works [3, 27] which tackle the problem of topic extraction from such problems. In these studies, topic extraction is used for grouping problems in terms of their related programming knowledge components, concepts or skills. For example, a problem that can be solved by using graph algorithms, such as breadth-first search, flood-fill or topological sort, can be classified into the graph category. Notice however that the target audience of these studies are more experienced POJ users. Instead, here we are not interested in categorising problems based on advanced topics.

In fact, we tackle, for the first time, to the best of our knowledge, the challenge of extracting the subject matter from programming problem statements available in POJ systems used in introductory programming, in order to improve the teaching and learning process of CS1 for non-CS courses, by matching problems to non-CS majors.

3. EDUCATIONAL CONTEXT

In this paper, we use as study base, as said, the CodeBench Online Judge environment, which is self-designed and implemented, as it allows us the freedom to add the changes inspired by our research results. Thus, we analyse here running the Introductory Programming (CS1) course at the Federal University of the Amazonas, via this self-designed POJ, which is delivered to 15 non-CS undergraduate degrees across the university. These courses are divided into 5 major areas: Mathematics, Physics, Engineering, Statistic and Geology. Three of the degrees belong to Mathematics, 2 to Physics, 8 to Engineering, 1 to Statistics and 1 to Geology. Figure 1 illustrates this configuration.

As Figure 2 illustrates, during the CS1 course, students in our environment typically solve 7 assignment lists with problems of increasing difficulty, using the Python programming language. They are allowed to solve the problems with an unlimited number of submission attempts, as long as they meet the deadline for solving all problems on a given list. The exercise lists always precede an exam on the same programming topic, both carried out in the Online Judge. Each list has an average of 10 questions, and the tests have 2 questions. We call a list together with its exam a ‘session’, where each session addresses a specific programming topic. Altogether, the course thus is formed of 7 sessions, that is, 7 programming topics are covered during CS1. Each session lasts on average 2 weeks.

During the 7 sessions, students work on the following programming topics: *Sequential*, *Composite conditional structures*, *Chained conditional structures*, *Repeating structures by condition*, *Repeating structures by count*, *Vectors* and *Strings* and *Matrices*. Before the 7 sessions, students have a

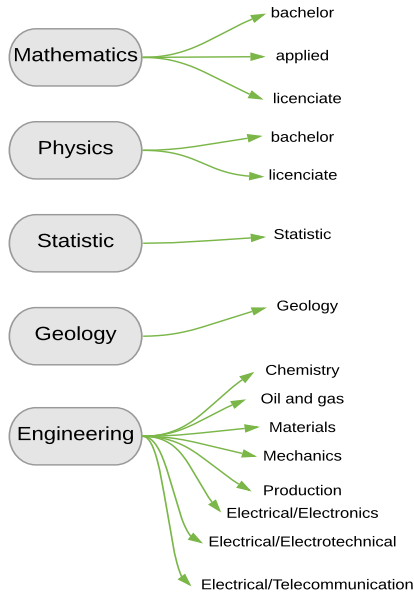


Figure 1: non-CS undergraduate courses at the Federal University of the Amazonas

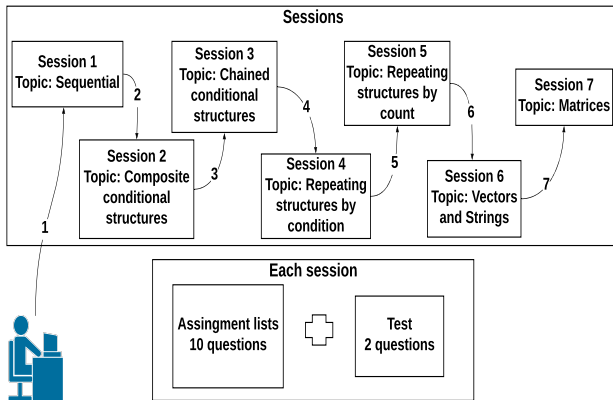


Figure 2: CS1 course configuration

first week to get used to the Python programming language, where they learn about *Variables* and *Single Operations*.

Whilst, in our online judge, problems are well structured based on these programming topics as above, they lack a clear division based on the contexts (here, related major areas) in which the problems are to be delivered. Please also note that, although the sessions are ordered by their increasing difficulty, the topics they are addressing are somewhat unrelated. Moreover, this increase in difficulty is typical for any CS1 course, be it offline or online.

Thus, our POJ is generic enough and is hence a good environment in which to research approaches to automatic classification by contexts, based on the statements, to build context-based personalised assignment lists, towards ul-

timately enhancing the engagement of non-CS students in their learning process.

4. DATA

The database in our Online Judge system consists of 986 programming problems in the CS1 discipline. As said, the statements in the database were initially not categorised by context; thus, we proceeded to create a labelled corpus, by manually classifying the contexts of each statement, to further use to carry out the experiments.

As labels, we adopted in this research contexts extracted from Zanini and Raabe’s definitions [26], which show that the context of problems plays an important role for novice programming students. Their study manually analysed the contexts of 428 programming problems statements used in introductory programming (as in our case) offered to 51 undergraduate courses. As a result, they found 20 possible contexts for these problems, as follows: mathematical, commercial, person, school, human resources, research, banking, physics, production, sport, computational, traffic, date and time, environment, tax, safety, consumption, population, others, and gamble.

We thus started with their proposed labels to annotate our problems. However, there were some groups of statements that could not be mapped over the above contexts. Moreover, the context “others” is too general and provides no real information. Given that, we removed the context “others” and propose here some additional contexts, as part of our contribution, in order to annotate our larger set of statements. As a result of the above process, we produced a total of 23 contexts, which we grouped together in a *new CS1 Context Taxonomy*, which is described in Table 1. This includes the following contexts, as contributions of our research: *Games, Movies and Series, Chemistry* and *Geography*. In addition, the table shows the number of statements for each context labelled and used in this research, the description of the contexts as well as the undergraduate courses that may have a high connection with the context.

It is worth noting that we performed a statistic test that measures inter-annotator agreement to validate if our annotation process was conducted properly. To do so, we used Cohen’s kappa (k) [4], which shows the level of agreement between two annotators on a classification task. As a result, we achieved a $k = 0.961$, which is considered almost perfect agreement [2].

5. METHODOLOGY

Figure 3 illustrates the proposed evaluation methodology pipeline used in the experiments of our research. We create here a unique, comprehensive pipeline, studying various combinations of the most popular and successful bleeding edge state-of-the-art techniques for natural language processing (NLP). The following subsections explain each step of our methodology.

5.1 Data augmentation

The data augmentation stage consists of balancing the training data by paraphrasing it, using the pre-trained model BERT [6]. Importantly for our task, this allows for *context-*

Context	Focus of the Statement	non-CS course	N
Mathematical	resolution of purely mathematical problems, without this being applied to another context	Mathematics and Engineering	261
Commercial	handling of products, goods, such as buying and selling, calculation of commission, provision of services	Economy	120
Games	game application, be it a virtual game or even a table game; for example, in the database there are games of naval battles, as well as video games	Digital games courses	96
School	to solve a school problem, such as averaging, passing or failing verification	Pedagogy	79
Traffic	related to the driver, car, mileage, accidents	All courses	43
Sport	some activity involved with sport, such as running, football, classification	Physical education	42
Physics	resolution of purely physical problems, without this being applied to another context	Physics and Engineering	36
Banking	related to bank transactions, investment, balance, withdrawal, deposit, stock exchange	Economy	35
Human Resources	problem related to human resources, such as salary calculation, data related to employees, calculation of bonuses, recruitment and selection of employees	Sociology and Psychology	35
Movies and TV Shows	problem situation in a film or TV shows. To illustrate, there are questions from the movie <i>Harry Potter</i> about potion calculation	All courses	30
Population	problems on population data, such as birth rate, mortality rate, population growth; referring to either human or animal population	Statistic	25
Chemistry	purely chemical problems, without this being applied to another context	Chemical engineering	23
Person	problems with elements directly related to a person, like weight, height, sex	All courses	22
Date and time	calculation of date or time, calculation of day, verification of month, conversion of hours, minutes and seconds, time interval	All courses	21
Safety	control access, password verification, data security, encryption, validation	Software engineering	20
Research	providing statistical data of opinion polls	Statistic and Journalism	18
Environment	relating to environmental issues, such as pollution, temperature	Environmental engineering	18
Health	related to issues of fighting diseases	Medicine	17
Consumption	calculation of water, electricity or telephone-related consumption	Economy	16
Geography	resolution of purely geographical problems, without this being applied to another context	Geology	11
Production	related to the production of products, the quantity produced, production value, origin of the products	Production engineering	7
Computational	computational issues, such as conversion of binary, decimal, hexadecimal numbers, ASCII table	Computer engineering	6
Tax	calculation of taxes, such as income tax	Economy	5

Table 1: Our proposed CS1 Context Taxonomy and Data Set description, with respective non-CS undergraduate course name and Number of items per Context, N

tual paraphrasing. Figure 4 illustrates a paraphrasing process based on a fragment of a statement from the category “Computational”.

Figure 4 shows a new generated sentence with clear semantics for a human reader. Still, generated text sometimes

misses such a clear structure. Nevertheless, our goal here is not to generate new sentences which could be meaningful for learners. Instead, we aim at creating artificial statements, which are not to be presented to humans, but will be used to expand the minority classes, providing variations to the predictive models (see bias-variance trade-off [8]). In other

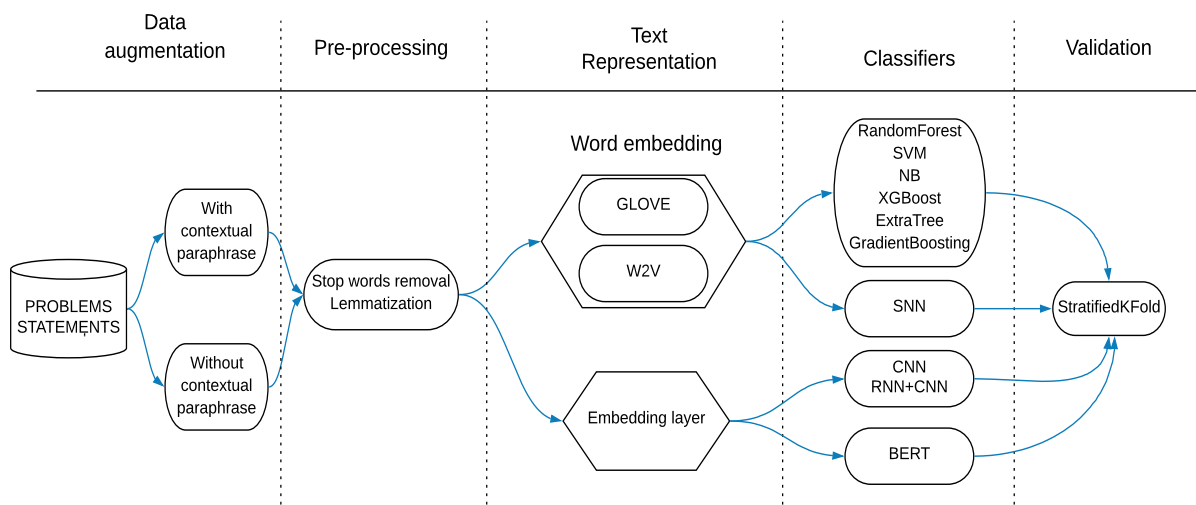


Figure 3: Proposed Automatic Contextualisation Research Methodology Pipeline

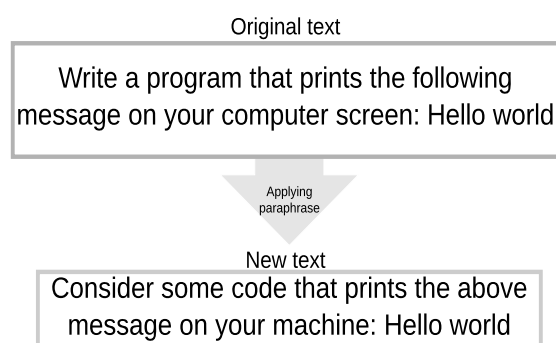


Figure 4: Paraphrasing Example using BERT [6]

words, despite this irregularity in the semantic sense of the statement, it is possible to perceive that the new instance generated belongs to the same class from which it was derived from and, therefore, it may represent a useful addition for the learning algorithm (which is later, as can be seen, confirmed by the results).

Nonetheless, as can be seen, despite the potential of such contextual paraphrasing, the new statements repeat some words from the original and keep almost the same number of tokens, which is a limitation of this method. As such, to prevent overtraining on artificial data (instances created using contextual paraphrases), we have set a limit of, at most, quadrupling the base of minority classes. We established this limit after some empirical experiments. That is, a statement is allowed to generate at most 4 new samples in the training base, as long as the new number of statements is below the number of instances of the majority class. Hence, this process may not render a perfectly balanced training base. To illustrate, imagine that the majority class has 10 questions on the training set, while the minority class has

1 question; with this paraphrasing algorithm, it is possible to extend the minority class for up to 5 questions (4 new samples + original statement).

In this work, experiments were carried out with and without paraphrasing, in order to analyse how the balancing by paraphrasing can influence the results.

5.2 Pre-processing

As we used reliable data (problems statements created directly by instructors/monitors), there was no need in our data processing of performing orthographic corrections, expanding contractions and other common data-cleaning steps. However, all our problem statements were originally in the Portuguese language. As there are many tools available for processing text written in English, we opted to translate our statements first into English, by using the *googleTrans*² library. Subsequently, we proceeded in applying our pipeline processing on the English text obtained, with and without the use of *stop-words* removal and *lemmatization*, using *spacy*³. As a result, we observed empirically that these two techniques were useful for data filtering in our pre-processing step. Next, we show how we further prepare the text for the machine learning algorithms.

5.3 Text Representation

The machine learning algorithms take as input a sequence of text to learn the structure of text, just like a human does. However, we need to convert the data in numerical form. As such, we represent our text data as a sequence of numbers (see Keras Tokenizer function⁴). Moreover, the ML algorithm expects each training instance to have the same length (same number of tokens). Thus we padded with zeros at the end sequences that are shorter than the maximum length

²pypi.org/project/googletrans/

³spacy.io

⁴keras.io/preprocessing/text/

sequence. To do so, we applied the Keras padding module⁵ over the sequences.

In addition, two different state-of-the-art NLP techniques for vector representation of words are used for competing against each other: googleNews-Vectors (W2V)⁶ and Glove [16] word embeddings. Moreover, for the BERT classifier, we used its own layer of word embeddings. Similarly, for the other deep learning models, we used the word embeddings layers as provided by the Keras library⁷. The purpose of this step is to compare the NLP techniques in terms of performance with our data set. Therefore, we created a *process to obtain the best model for automatic categorisation of contexts of programming questions* for our educational context. The process allowed us thus to carry out experiments with advanced Deep Learning methods, and to compare not only those approaches with each other, but also with classical approaches, such as shallow learning models.

5.4 Classifiers

For deep learning models we used: a) Convolutional Neural Networks (CNN) which have a convolutional layer, followed by three dense layers; b) Recurrent Neural Networks (RNN), with a recurring layer using a Long Term and Short term memory (LSTM) followed by three dense layers; c) RNN and CNN (RNN+CNN) stacked with the same configurations as those of the items a and b; d) Sequential Neural Network (SNN) with two dense layers and e) BERT for classification (notice that we used BERT for two purposes: i) perform contextual paraphrasing; ii) multi-classification).

As we are tackling a multi-classification problem, the final layer for each neural network was represented by a softmax layer [13]. For all deep learning models, the configurations used above represent the default recommended ones from the literature [13].

Additionally, we used the following classical, shallow classifiers, with the word embeddings from googleNews-Vectors and Glove: Random Forest Classifier (RFC), Support Vector Machine (SVM), Extremely Randomised Tree Classifier (ETC), Gaussian Naive Bayes (GNB), XGBoost (XGB) and Gradient Boosting Classifier (GBC).

5.5 Validation

To validate the models, we employed the stratified validation with 10 *folds*. This method divides the base into k partitions, using $k - 1$ for training and 1 for testing. After that, the accuracy of the test partition is calculated. This process is repeated k times, until all partitions have been used as a test. Finally, the average of the accuracy obtained in the tests is computed. It is noteworthy that each *fold* was divided proportionally to the number of statements present in each class in the database [13]. We implemented it using the *StratifiedKfold* from scikit-learn. Notice that we performed the data augmentation only on the training sets of each training fold. Thus, there were no paraphrased texts in the test sets.

⁵keras.io/preprocessing/sequence/

⁶code.google.com/archive/p/word2vec/

⁷keras.io/

To evaluate our models, we used the F1-score, as this metric combines precision and recall in an harmonic mean. This is useful because it gives much more weight to low values than a regular mean, which treats all values equally. Moreover, we used the weighted F1-score, which takes into account the proportion of each class.

6. RESULTS AND DISCUSSION

We built a total of 34 predictive models. Figure 5 illustrates all the results obtained by all models applied in this research. From this figure, we can notice that paraphrasing improved the (weighted) *F1-score* in all models. To illustrate this boosting, the model *GLOVE + SVM* achieved a F1-score of 86%, without paraphrasing. Whereas with the paraphrasing, the model achieved 94%, an increase of 8%. To validate that, we performed the McNemar's hypothesis statistical test, which is recommended to compare machine learning models [7]. We compared the models with or without the contextual paraphrasing. As a result, we confirmed that the paraphrasing statistically boosts all models, even after Bonferroni correction ($p - values \ll 0.05/2$). Table 2 shows the classification performance of the models in terms of macro and weighted precision, recall and F1-score. Moreover, this table shows the accuracy of each model.

From a visual inspection of Figure 5, we can argue that the best model found is the BERT classifier with use of the contextual paraphrasing (BERT + PAR), as the model has the highest median and a low standard deviation. Moreover, this model achieved the highest recall, F1-score and accuracy (Table 2). To validate that, we also performed McNemar's test. As a result, we confirmed our previous deduction as *BERT + PAR* statistically outperforms all the other models, even after Bonferroni correction ($p - values \ll 0.05/33$). As such, in Figure 6, we show the performance of this model for each context, as a *heat-map plot*. The rows represent the actual values, while the columns depict the predicted contexts.

Figure 6 illustrates that, in general, our best model is capable of recognising problems from each context with a high recall. Indeed, there are predictions in some classes without miss-classification such as *Computational*, *Sports*, etc. However, we can see some cases where the model made mistakes. For example, the model gets confused between the classes *Production* and *Commercial*. This may have happened because some problem statements could have come from a production context, but with focus on sales, which would be further related to the *Commercial* context. Moreover, there are some problems that are actually from the context *Production*, classified by our best model as *Date and time*. This was an unexpected result for us. After visual inspection, we noticed that some of these problems linked the efficiency of a company to the time-scale (e.g., how long a process took determined its efficiency). This is a possible explanation for such confusions within our model.

Coupled with that, according to Table 1, it is possible to notice that the class *Computational* has only a few statements. Despite this low number of problems in this context, our model is able to recognize this minority class with no errors (100% of precision and recall). Still, the class *Tax* presents the lowest number of problems in our database.

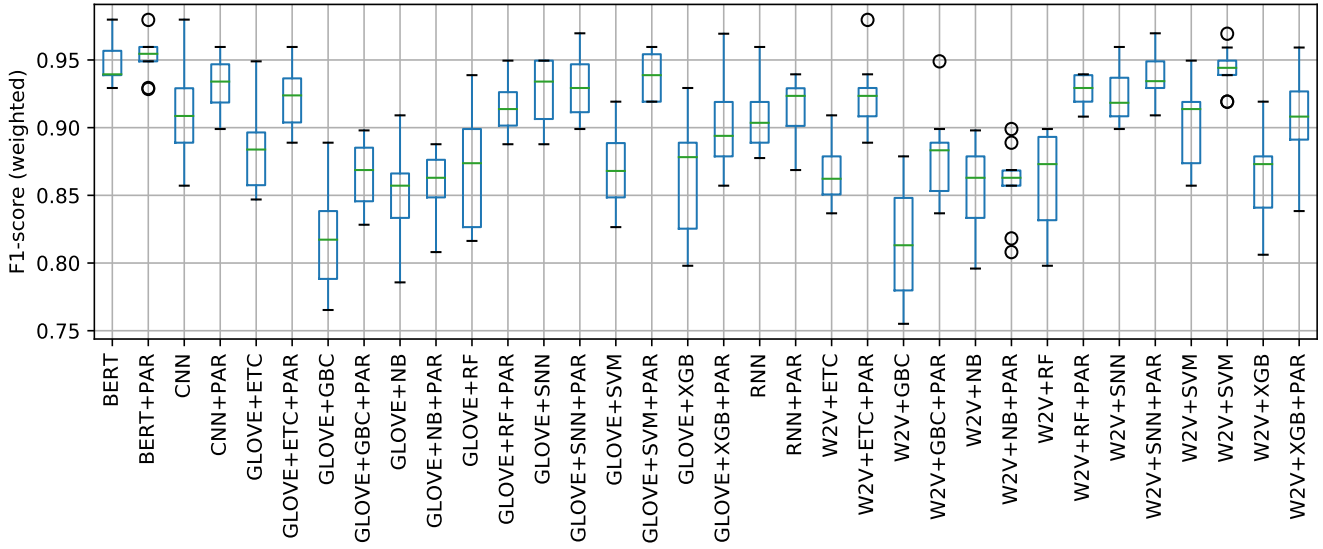


Figure 5: All results (F1-score)

But even so, our model achieved a recall of 80% in this context. Consider that the instances missclassified from the *Tax* context were allocated to *Commercial*, which makes sense, as, in some cases, these two contexts are related.

Although the model achieved a high recall (95%) in the context of *Games*, instances that the model was not able to recognise were spread through multiple contexts (*Commercial*, *Date and time*, *Physics*, *Tax*, and *Mathematical*). The 2% error between *Games* and *Tax* can be explained by statements of games that comprise tariffs, e.g., when buying a certain product within the game. For example, there are statements in our data set that discuss buying products for a character, such as a battle suit. Further, the error of 1% with the class *Commercial* could be due to a reason similar to that of the class *Tax*. To illustrate, within a game, some statements comprise the purchase of products. Regarding the class *Date and time*, an explanation would be statements that address some mission that the character needs to accomplish in a specific time. Regarding the error in the classes *Physics* and *Mathematical*, it may be due to statements in games that contain speed calculation.

Another important analysis to be done occurs in the class *Research*. The model achieved a recall of 94%, whereas 6% of errors occurred in the class *Person*. One possible reason is that surveys are conducted based on a group of people. Also, there are statements in our database that contain research carried out on some characteristics of people, such as age group, education, etc.

Another interesting outcome relates to the following classes: *Banking* and *Commercial*. Note that both presented confusion errors between each other, that is, the class *Commercial* presented wrong predictions in the class *Commercial* and vice-versa. This is justified because both classes deal with statements that involve money.

Furthermore, a similar situation occurs for the classes *Health* and *Population*. Here, errors could be due to statements addressing, e.g., the growth of a virus or bacteria. Thus, results may highlight relations between these contexts.

Another interesting analysis relates to the majority class of our data set, that is, the class *Mathematical*. Note that it was possible to obtain here a 99% recall. Even more importantly, note that few classes have errors in this class, that is, although we are dealing with the majority class, our model can differentiate, with high precision, all classes, against this one. To illustrate, only the following classes had a confusion error with respect to this class: *Games*, *Geography* and *Commercial*. Regarding the error presented in the prediction of the class *Games*, it is an error that could be justified by questions that deal with any type of calculation, given that any form of calculation can be directly related to the mathematical context. For the *Geography* class, the error could be justified, as we have noticed the existence of statements that deal with map scale conversion. Regarding the class *Commercial*, the error could be justified by calculating the price of a certain product.

Nevertheless, we had unexpected outcomes as well. For example, it was arguably to be expected that the *Physics* class presented errors in the *Mathematical* class, given that statements that address a physical contextualisation deal with mathematical calculations. However, this does not happen. Thus, our model clearly differentiates here between even small details present in the statement of each context.

In other words, although there is an error in the classification of some instances in the classes, most of these errors can be easily justified. This may suggest that the statements worked on in this research have multi-contextualisation, that is, a statement can address more than one context. However, what happens in practice is that one context is predominant, and the prediction of our model reflects this. Still, it is

Table 2: Classification performance of the predictive models (Pr: precision; Re: recall; F1: F1-score; Acc: accuracy).

Model	Pr(Macro)	Pr(weighted)	Re(Macro)	Re(weighted)	F1(Macro)	F1(weighted)	Acc
GLOVE+RFC	95%	88%	78%	87%	84%	87%	86.8%
GLOVE+RFC + PAR	94%	92%	86%	92%	89%	91%	91.6%
GLOVE+ETC	95%	90%	80%	88%	86%	88%	88.3%
GLOVE+ETC+PAR	95%	93%	87%	92%	90%	92%	92.3%
GLOVE+XGBC	91%	87%	74%	87%	79%	86%	86.5%
GLOVE+XGBC+PAR	92%	91%	82%	90%	85%	90%	90.2%
GLOVE+GNB	90%	86%	78%	85%	82%	85%	85.0%
GLOVE+GNB + PAR	88%	87%	82%	86%	84%	86%	86.7%
GLOVE+SVM	80%	87%	71%	87%	75%	86%	86.8%
GLOVE+SVM+PAR	95%	94%	89%	94%	91%	94%	93.7%
GLOVE+GBC	79%	83%	68%	82%	72%	81%	81.7%
GLOVE+GBC+PAR	83%	87%	77%	87%	79%	86%	86.6%
GLOVE+KC	91%	93%	89%	93%	90%	93%	92.8%
GLOVE+KC+PAR	91%	93%	90%	93%	90%	93%	93.1%
W2V+RFC	95%	88%	78%	86%	84%	86%	86.1%
W2V+RFC+PAR	94%	93%	87%	93%	90%	93%	92.7%
W2V+ETC	95%	88%	79%	87%	85%	87%	86.8%
W2V+ETC+PAR	95%	93%	87%	92%	90%	92%	92.3%
W2V+XGBC	92%	87%	76%	86%	82%	86%	86.4%
W2V+XGBC+PAR	91%	91%	86%	91%	87%	91%	90.7%
W2V+GNB	90%	87%	78%	86%	82%	86%	85.7%
W2V+GNB+PAR	88%	87%	81%	86%	84%	86%	85.9%
W2V+SVM	85%	90%	79%	90%	82%	90%	90.2%
W2V+SVM+PAR	96%	95%	91%	94%	93%	94%	94.3%
W2V+GBC	77%	82%	69%	81%	73%	81%	81.3%
W2V+GBC+PAR	83%	88%	78%	88%	80%	88%	87.8%
W2V+KC	91%	92%	89%	92%	90%	92%	92.4%
W2V+KC+PAR	93%	94%	91%	94%	92%	94%	93.9%
KT+CNN	94%	91%	84%	91%	88%	91%	90.8%
KT+CNN+PAR	92%	93%	90%	93%	91%	93%	93.2%
KT+(RNN+CNN)	86%	91%	86%	91%	85%	91%	90.8%
KT+(RNN+CNN)+PAR	89%	91%	87%	91%	88%	91%	91.4%
BT+BERT	93%	95%	91%	95%	92%	95%	94.7%
BT+BERT+PAR	94%	95%	92%	95%	93%	95%	95.2%

potentially useful to further analyse this problem as a multi-contextual prediction task.

7. LIMITATIONS

One of the major limitations of this paper is related to data set size. Although we have a significant number of problems, in the case of some contexts there is a small number of instances, due to the quantity of classes in our multi-classification problem. To address this limitation, we used cutting-edge NLP techniques to produce new instances on the training set, using contextual paraphrases.

Moreover, our original problem descriptions were in Portuguese and hence, when we translated them to English, this may have introduced some errors from our automatic data processing. However, this was counter-balanced by the availability of the most cutting-edge NLP processing tools for the various steps involved in our pipeline, which were not available for the Portuguese language.

In addition, this research worked with introductory topics to computer programming. It is thus less clear if the methodology applies to more advanced topics of programming. For example, database disciplines may need a different approach. However, the holistic pipeline we propose can guarantee that the right method can outperform the others, thus ensuring area appropriateness.

Another limitation arises from undergraduate courses that do not have programming in their curriculum. Although it is clear that in this research several courses may use programming for some activities, not all of them have programming topics in the curriculum. To illustrate, although our data set presents health issues that can be applied to the medical or nursing courses, unfortunately these undergraduate courses do not have programming topics in their curriculum. This may however change in the future, with the rise of the ubiquitousness of computing, and thus this research may have wider relevance and impact than originally envisioned.

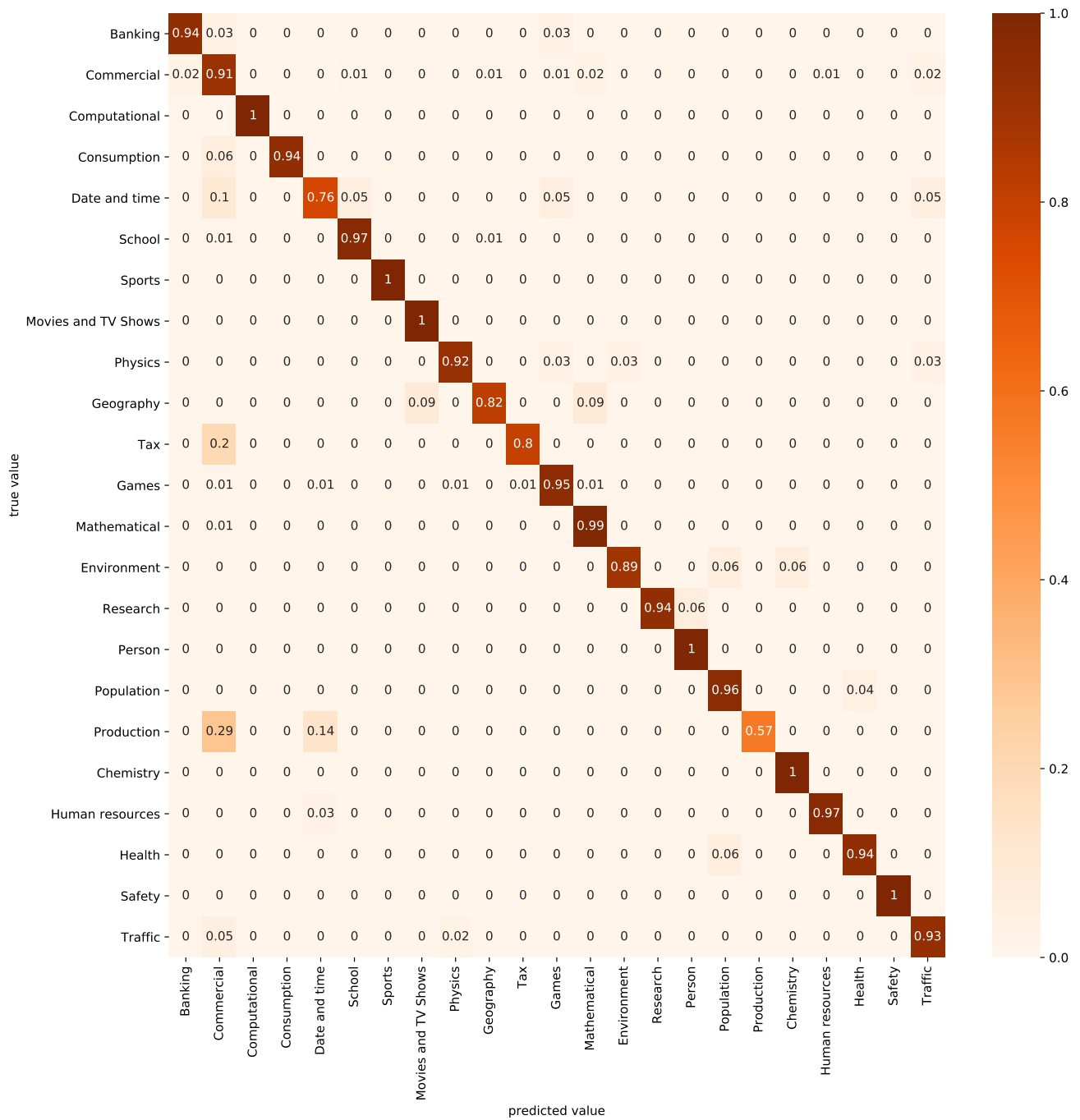


Figure 6: BERT with Paraphrasing

8. CONCLUSION AND FUTURE WORKS

According to the results obtained and illustrated in this research, we can conclude that paraphrasing of the minority classes boosts results, that is, it was able to make predictive models more accurate and with greater recognition capacity, regardless of which NLP was used, that is, Glove, Word2vec, BERT, etc.

In addition, our work was able to achieve a performance with high precision and a high recognition rate for all 23 classes

proposed in this article. That is, our best model, which is based on the BERT technique with paraphrase-balancing, was able to achieve an accuracy of 95.2% with a minimal error rate, which is no more than 4.8%.

With that, the first step to *generate personalised problem lists, according to the context of the undergraduate course*, was taken. We have additionally provided a *new context taxonomy for problems*, as well as a *comprehensive evaluation pipeline methodology for context-based personalisation*

of problem lists.

As future work we intend to further evaluate the effect of the personalised programming problem assignments using our method to detect the subject matter. Thus, we can explore if the performance of the non-CS students will be affected when solving problems related to their courses.

In addition, three new experiments can be performed to analyse the generalisation power of our method. The first is to repeat the procedure on other online judge problem collections, but still at an introductory programming discipline level. The purpose of this experiment is to verify how generalisable our approach is across educational settings different from ours. We believe, nevertheless, that choices such as the programming language used in teaching CS1 will not be a factor that will prevent similar outcomes.

As a second experiment, we would repeat the procedure with more advanced programming topics, to analyse if the method can be applied to these more complex types of topics. For example, disciplines such as data structures may be a research target. Finally, we envision to adapt our pipeline to perform automatic classification of the programming problems in terms of the topics used in the CS1 courses (*Sequential, Composite conditional structures, Chained conditional structures, Repeating structures by condition, Repeating structures by counting, Vectors and Strings and Matrices*). Such a pipeline would be useful for several applications, such as for problem recommendation, automatic annotation, amongst others.

Concluding, we believe that the automatisisation of the classification of statements by contexts is extremely relevant for several reasons, among which we highlight: i) statements which students are already familiar with can help in the process of engagement and learning; ii) students will find it easier to understand the relevance of programming in their professional lives; iii) teachers can use this automatisisation to generate personalised lists, which would facilitate their work, since it would be too much work to select these problems manually, in addition to which it could lead to human error and iv) students could use this automatisisation to select problems to which they are used to, facilitating their process of learning a certain programming topic.

Acknowledgements

This research, carried out within the scope of the Project for Education and Research (SUPER), according to Article 48 of Decree n° 6.008/2006 (SUFRAMA), was partially funded by Samsung Electronics of Amazonia Ltda, under the terms of Federal Law n° 8.387/1991, through agreements 001/2020 and 003/2018, signed with Federal University of Amazonas and FAEPI, Brazil.

9. REFERENCES

- [1] T. Aljohani, F. D. Pereira, A. I. Cristea, and H. T. Oliveira. Prediction of users' professional profile in moocs only by utilising learners' written texts. In *International Conference on Intelligent Tutoring Systems*. Springer, 2020.
- [2] R. Artstein and M. Poesio. *Inter-coder agreement for computational linguistics*. Educational and Psychological Measurement 20(1):37-46, 2008.
- [3] V. Athavale, A. Naik, R. Vanjape, and M. Shrivastava. Predicting algorithm classes for programming word problems. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 84–93, 2019.
- [4] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20(1):37-46, 1960.
- [5] R. E. De Castilho, J.-C. Klie, N. Kumar, B. Boullosa, and I. Gurevych. Linking text and knowledge using the inception annotation platform. In *2018 IEEE 14th International Conference on e-Science (e-Science)*, pages 327–328. IEEE, 2018.
- [6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [7] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923, 1998.
- [8] P. Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.
- [9] F. Dwan, E. Oliveira, and D. Fernandes. Predição de zona de aprendizagem de alunos de introdução à programação em ambientes de correção automática de código. *Simpósio Brasileiro de Informática na Educação-SBIE*, 28(1):1507, 2017.
- [10] L. Echeverría, R. Cobos, L. Machuca, and I. Claros. Using collaborative learning scenarios to teach programming to non-cs majors. *Computer Applications in Engineering Education*, 25(5):719–731, 2017.
- [11] S. Fonseca, E. Oliveira, F. Pereira, D. Fernandes, and L. S. G. de Carvalho. Adaptação de um método preditivo para inferir o desempenho de alunos de programação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 30, page 1651, 2019.
- [12] L. Galvão, D. Fernandes, and B. Gadelha. Juiz online como ferramenta de apoio a uma metodologia de ensino híbrido em programação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 27, page 140, 2016.
- [13] A. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2019.
- [14] I. Govender. The learning context: Influence on learning to program. *Computers & Education*, 53(4):1218–1230, 2009.
- [15] V. T. Norman and J. C. Adams. Improving non-cs major performance in cs1. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education, SIGCSE '15*, page 558–562, New York, NY, USA, 2015. Association for Computing Machinery.
- [16] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

- [17] F. Pereira, E. Oliveira, D. Fernandes, L. S. G. de Carvalho, and H. Junior. Otimização e automação da predição precoce do desempenho de alunos que utilizam juízes online: uma abordagem com algoritmo genético. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 30, page 1451, 2019.
- [18] F. D. Pereira, E. Oliveira, A. Cristea, D. Fernandes, L. Silva, G. Aguiar, A. Alamri, and M. Alshehri. Early dropout prediction for programming courses supported by online judges. In *International Conference on Artificial Intelligence in Education*, pages 67–72. Springer, 2019.
- [19] F. D. Pereira, E. H. Oliveira, D. Fernandes, and A. Cristea. Early performance prediction for cs1 course students using a combination of machine learning and an evolutionary algorithm. In *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, volume 2161, pages 183–184. IEEE, 2019.
- [20] F. D. Pereira, E. H. T. Oliveira, D. B. F. Oliveira, A. I. Cristea, L. S. G. Carvalho, S. C. Fonseca, A. Toda, and S. Isotani. Using learning analytics in the amazonas: understanding students’ behaviour in introductory programming. *British journal of educational technology.*, 2020.
- [21] Y. Qian, S. Hambruch, A. Yadav, and S. Gretter. Who needs what: Recommendations for designing effective online professional development for computer science teachers. *Journal of Research on Technology in Education*, 50(2):164–181, 2018.
- [22] J. M. C. RAABE, A. L. A.; SILVA. Um ambiente para atendimento as dificuldades de aprendizagem de algoritmos. pages 2326–2335, 2005.
- [23] B. L. Santana and R. A. Bittencourt. Increasing motivation of cs1 non-majors through an approach contextualized by games and media. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–9, Oct 2018.
- [24] A. Stefik and S. Siebert. An empirical investigation into programming language syntax. *ACM Transactions on Computing Education (TOCE)*, 13(4):1–40, 2013.
- [25] S. Wasik, M. Antczak, J. Badura, A. Laskowski, and T. Sternal. A survey on online judge systems and their applications. *ACM Computing Surveys (CSUR)*, 51(1):1–34, 2018.
- [26] A. S. Zanini and A. L. A. Raabe. Análise dos enunciados utilizados nos problemas de programação introdutória em cursos de ciência da computação no brasil. In *Anais do XXXII Congresso da Sociedade Brasileira de Computação, XX WEI – Workshop sobre Educação em Computação, Curitiba*, 2012.
- [27] W. X. Zhao, W. Zhang, Y. He, X. Xie, and J.-R. Wen. Automatically learning topics and difficulty levels of problems in online judge systems. *ACM Transactions on Information Systems (TOIS)*, 36(3):1–33, 2018.

Enhancing Student Competency Models for Game-Based Learning with a Hybrid Stealth Assessment Framework

Nathan Henderson¹, Vikram Kumaran¹, Wookhee Min¹, Bradford Mott¹, Ziwei Wu¹,
Danielle Boulden¹, Trudi Lord², Frieda Reichsman², Chad Dorsey², Eric Wiebe¹, James Lester¹

¹North Carolina State University
Raleigh, NC, 27695

{nlhender, vkumara, wmin, bwmott, zwu17, dmboulde, wiebe, lester}@ncsu.edu

²Concord Consortium
Concord, MA, 01742
{tlord, freichsman, cdorsey}@concord.org

ABSTRACT

In recent years, game-based learning has shown significant promise for creating engaging and effective learning experiences. Developing models that can predict whether students will struggle with mastering certain concepts could guide adaptive support to assist students with mastering those concepts. Game-based learning environments offer significant potential for unobtrusively assessing student learning without interfering with gameplay through stealth assessment. Prior work on stealth assessment has focused on a single machine learning technique such as dynamic Bayesian networks or long short-term memory networks; however, a single modeling technique often does not guarantee the best predictive performance for all concepts of interest. In this paper, we present a hybrid data-driven approach to stealth assessment for predicting students' mastery of concepts through interactions with a game-based learning environment for introductory genetics. Stealth assessment models utilize students' observed gameplay behaviors using challenge- and session-based features to predict students' learning outcomes on identified concepts. We present single-task and multi-task models for predicting students' mastery of concepts and the results suggest that the hybrid stealth assessment framework outperforms individual models and holds significant potential for predicting student competencies.

Keywords

Stealth Assessment, Predictive Student Modeling, Game-based Learning, Multi-Task Learning

1. INTRODUCTION

Recent years have seen growing interest in game-based learning environments because of their potential for creating engaging and effective learning experiences [7, 43]. Researchers have investigated game-based learning environments in a wide array of

domains, including mathematics [13, 40], computational thinking [4, 16], and science [2, 10, 30].

While a common gameplay design adopted by many game-based learning environments is providing students with a fixed sequence of levels with increasingly difficult challenges per concept, game-based learning environments could provide individualized sequences of challenges and just-in-time support, so that students can focus on gameplay at the edge of their knowledge and skills and remain engaged throughout the learning experience [20, 44]. To achieve this goal, game-based learning environments should be equipped with the ability to detect when students are struggling or have gaps in their knowledge and take appropriate action to tailor their learning experience [33]. Presenting in-game challenges adaptively tailored to individual students' knowledge can play a crucial role in supporting mastery learning and promoting engagement while effectively addressing problems with a one-size-fits-all approach.

With recent advances in machine learning, data-driven approaches using students' in-game behaviors have enabled the automatic assessment of students' evolving competence [1, 21] and the modeling of mind wandering [19], wheel spinning [25], and quitting behaviors [12], all of which are associated with negative learning outcomes. A robust modeling of student behaviors can guide students from undertaking a challenge that is beyond their capabilities as well as facilitate their engagement through individualized learning activities tailored to their competencies for knowledge and game-playing skills.

There is now a sizable literature on stealth assessment in game-based learning [34]. Stealth assessment robustly measures student learning without disrupting engagement by embedding unobtrusive assessments within game mechanics, offering real-time non-disruptive assessment [35]. Building on evidence-centered design (ECD) [24], which provides a systematic approach to developing knowledge assessments, stealth assessment examines student interaction data (i.e., evidence model) with in-game challenges (i.e., task model) to provide real-time behind-the-scenes measurement of students' learning processes and outcomes (i.e., competency model) [22, 36]. Specifically, students' learning is inferred by analyzing low-level sequences of observed problem-solving behaviors that manifest competencies for knowledge and skills without conducting explicit formative assessments. Inferences made by stealth assessment models can inform effective

Nathan Henderson, Vikram Kumara, Wookhee Min, Bradford Mott, Ziwei Wu, Danielle Boulden, Trudi Lord, Frieda Reichsman, Chad Dorsey, Eric Wiebe and James Lester "Enhancing Student Competency Models for Game-Based Learning with a Hybrid Stealth Assessment Framework" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 92 - 103

scaffolding strategies (e.g., adaptive challenge selection, tailored problem-solving support) for individual learners in a timely and contextually appropriate manner [29, 36]. It can also guide teachers to potential pedagogical adaptations or support integration with additional curricular activities, which are core components of distributed and integrated scaffolding [26, 28, 39].

In this work, we investigate stealth assessment with *Geniventure*, a game-based learning environment for introductory high school genetics learning. We present modeling approaches including single-task and multi-task random forest and recurrent neural network-based regression models for predicting students' competencies, whose labels were derived from students' post-test scores on genetics-focused concepts. In contrast to much previous work on stealth assessment that used a single machine learning technique, we present a hybrid stealth assessment framework that effectively leverages predictive capacities of all the explored modeling approaches. We compare the models' fitness to the data to gain insight into which combinations of models perform optimally across all the concepts, as well as which models are effective for individual concepts. The hybrid stealth assessment framework outperforms individual computational techniques with respect to predictive performance for student concept-level competencies.

2. RELATED WORK

Intelligent game-based learning environments simultaneously leverage capabilities of digital games to motivate students' learning through engaging narratives, virtual environments and intelligent tutoring systems (ITSs) to foster students' learning through adaptive scaffolding and context-sensitive feedback [15]. These environments facilitate learning through individualized challenges, narratives, feedback, and problem-solving support [30, 35, 42]. Students' fine-grained, sequential game trace data has been used in a wide range of student modeling tasks such as inferring the level of competency [22, 35], predicting affective states [3, 31], and recognizing students' learning goals [23]. In comparison to single-task learning investigated in much of previous student modeling work, recent years have seen a growing interest in the use of multi-task learning, a regularization method that exploits commonalities and differences across related tasks for improved generalizability. Multi-task learning has been examined for various student modeling tasks such as predicting student competencies in programming in a massive open online course (MOOC) [27] and modeling student performance in a game-based learning environment for middle-grade microbiology education [9], which demonstrated improved predictive performance relative to the single-task modeling approach. Similarly, Chaudhry et al. used multi-task modeling with both hint usage and knowledge tracing to induce models of students using online tutoring systems [5].

Stealth assessment is methodologically grounded in evidence-centered design (ECD), which was proposed to construct educational assessments in terms of evidentiary arguments [24]. ECD features task, evidence, and competency models to conduct probabilistic reasoning about knowledge, skills, and abilities of students utilizing evidence captured from interactions with learning tasks. Stealth assessment conducts real-time processing of data derived from these three ECD models that informs intelligent, adaptive game-based learning environments through devising robust evidence and competency models as well as creating task models that effectively develop the competencies [20]. While human expert-designed Bayesian networks have been examined as the core computational method for both competency and evidence

models for stealth assessment [37], another body of work has investigated an assessment pipeline that does not require costly domain knowledge engineering. Falakmasir et al. investigate the use of hidden Markov models (HMMs) to model student proficiency within educational games [8]. The log-likelihoods are approximated by the HMMs using sequential gameplay data, with the difference between the likelihoods serving as the independent variable for post-test prediction models. The authors of this work use linear regression to predict the student's post-test scores. There has also been growing interest in deep neural network architectures due to their capability to learn salient features from low-level, sequential data captured from interactions with task models [1, 20]. Long short-term memory network-based stealth assessment models have demonstrated significant promise by outperforming competitive baselines with respect to predictive performance of inferring students' competencies, while effectively eliminating the need to manually craft evidence rules and evidence models. In contrast to much of previous research, our work presents a hybrid stealth assessment framework that utilizes a suite of competency models to optimally harness distinguished predictive capacity yielded by a range of single-task and multi-task stealth assessment models.

3. DATASET

3.1 *Geniventure* Learning Environment

To evaluate the performance of our hybrid stealth assessment framework, we use gameplay interaction log data collected from students engaged with a game-based learning environment for introductory genetics for middle school and high school students (students ages 11-18 years), *Geniventure*. The design of the game is guided by core genetics-based concepts that align with the Next Generation Science Standards [38]. *Geniventure* engages students in exploring heredity, dominant and recessive traits, and the protein-to-trait relationship by breeding and studying drakes, a model species for dragons [18].

The game consists of 60 increasingly difficult puzzle-like challenges across 6 levels (Figure 1). Each of the challenges is part of a "mission", with each level containing multiple missions. The genetics concepts that the game addresses are presented through a variety of challenge types. While the game was designed to be played through in a linear fashion, students have the freedom to attempt challenges at any level and are allowed to quit a challenge at any time.

In the first half of the game, students are asked to change the drake's genotype to match a target phenotype (Figure 1, Level 1). To successfully complete these problem-solving challenges, students must understand several genetic concepts and be able to infer the phenotype of their drake from its genotype. Once students feel they have the correct genotype, they click the "Check" or "Hatch" button to submit their answer. If the drake they create matches the target drake, the challenge is successfully completed. Otherwise, the game provides the student with three progressively more directed levels of hints, as well as a visual cue, and allows them to continue to make further changes to the alleles until they quit or successfully complete the challenge. This model of counting moves and giving feedback in the form of hints is carried through the subsequent levels of the game, even though the challenge types vary. Other challenges instruct the user to match a phenotype to a given genotype, following a reversed procedure from Level 1 (Figure 1, Level 2), and also introduce scale color and other additional complexities to the challenges (Figure 1, Level 3).



Figure 1. Example challenges in *Geniventure* for the six gameplay levels.

The latter half of the game introduces more difficult concepts such as breeding and inheritance. Through several scaffolded challenges, students breed parent drakes with the goal of matching target offspring (Figure 1, Level 4). The tasks grow increasingly complex as students progress through this level, eventually culminating in a challenge requiring students to breed two parents to produce offspring that match a given drake. Students are also introduced to test cross, a genetic method for determining the genotype of one organism by crossing it with a fully recessive organism (Figure 1, Level 5). Finally, the last level introduces traits with more complex inheritance patterns, such as X-linked and polyallelic traits (Figure 1, Level 6). This level contains challenges illustrating concepts from all of the preceding levels such as allele target match, egg drop, meiosis, breeding, and test cross.

As previously mentioned, students can validate their work at any time and are provided with system-generated hints based on their perceived understanding of the genetics concepts if necessary. Hint usage, as well as time spent on challenges, and the students' success rate during their respective gameplay sessions, serves as the foundation for the features used to train the competency models.

3.2 Data Collection

The dataset was collected from 462 students from seven high schools and one middle school located in the Middle to Northern Atlantic coast of the United States. This data was collected during a teacher-led classroom implementation of *Geniventure* where students played the game during class over the course of several days. During gameplay, students' gameplay trajectory and their detailed in-game actions were recorded as trace data logs. Before playing the game, students took a pre-test consisting of 28 questions related to the genetic concepts covered in the game. Once gameplay concluded, students took a post-test which was identical to the pre-test (Figure 2). This assessment was aligned to the ECD competency model of the game and previously validated through two rounds of expert review and cognitive interviews with students. In administration, it demonstrated an internal consistency reliability of $\alpha = 0.873$. Both the pre-test and post-test were online surveys

accessible through the same online portal as the game. 38 students were removed due to the partial or missing pre/post test data. 108 students were removed due to missing trace data, resulting in a dataset containing trace data from 316 students. Results from a paired t-test on students' knowledge pre-test ($M = 14.41$, $SD = 5.826$) and post-test ($M = 19.33$, $SD = 6.131$) revealed a significant improvement from pre-test to post-test ($t(315) = 14.663$, $p < 0.01$, Cohen's $d = 0.823$). A majority of the students attempted between 50 and 150 challenges. The fewest number of challenges attempted by a student was 5, which serves as the basis for the sequence length of the subsampling window used to generate the sequential data for the competency models. The most challenges attempted by a student during the duration of the study is 248. To further illustrate the distribution of the number of challenges attempted per student, a histogram of the students' gameplay trajectories is shown in Figure 3.

4. ECD FOR STEALTH ASSESSMENT

Evidence-centered design (ECD) is a systematic approach for designing and developing reliable knowledge assessments in terms of evidentiary arguments [24]. When utilized to identify and analyze user behavior in online learning environments, it serves as the basis of stealth assessment in game-based learning environments [34]. While historically ECD has been utilized in the development of summative assessments, recent years have seen its application in the design of formative stealth assessment models for game-based learning environments [20, 34]. Assessment results inferred by stealth assessment models can be utilized to support student learning through adaptive scaffolding within the learning environment and also inform teachers about student learning trajectories through a teacher dashboard. As noted above, stealth assessment is grounded in three core ECD models. These three models were applied to the current study using *Geniventure* as follows:


- **Task Model:** This model defines the activities, or tasks, that students undertake as part of their learning. In the *Geniventure* learning environment, the task model consisted of 60 challenges across six game levels that students undertake.

These tasks focus on genetics concepts such as heredity, dominance/recessive, and the protein-to-trait relationship.


- **Evidence Model:** The evidence model takes as input low-level action sequences students produce while interacting with the game-based learning environment. Game-based learner behaviors are linked to targeted concepts to generate machine-interpretable evidence that can be directly utilized with the modeling techniques presented here. That is, a probabilistic model is constructed from analysis of a series of actions related to mastery (or not) of a particular concept. The evidence model informs the competency model in order to update its belief of students' competencies as they interact with the tasks.
- **Competency Model:** Mastery of 16 concepts (Table 1) are dynamically estimated by the competency model with respect to students' genetics knowledge. The concepts were derived from expert review of classroom learning goals and state science standards. The ground truth for their summative competencies are acquired from students' post-test scores on an explicit content knowledge assessment. The competency model is aligned to the summative post-test through the same set of ECD-derived concepts in Table 1.

In training the stealth assessment models, we extract competency scores based on correctness of students' individual responses to items on a post-test knowledge assessment (Figure 2). Competencies for a single concept in our competency model can be

Here are two whiptail parents. Remember, straight wings (W) are dominant to curly wings (w), fire breathing (F) is dominant to non-fire breathing (f) and dark gray (G) is dominant to light gray (g).



Dad
(Ww, Ff, gg)



Mom
(Ww, FF, Gg)

Question #8

Which combination of gametes can these parents produce that would combine to create a curly-winged, fire breathing, dark gray whiptail baby?

☐ Dad (w,F,g) and Mom (w,f,g)
☐ Dad (W,F,G) and Mom (w,f,g)
☐ Dad (w,F,g) and Mom (W,f,g)
☐ Dad (w,f,g) and Mom (w,F,G)

Question #9

Which combination of gametes can these parents produce that would combine to create a straight-winged, fire-breathing, light gray whiptail baby?

☐ Dad (W,f,g) and Mom (W,f,g)
☐ Dad (W,f,g) and Mom (w,F,g)
☐ Dad (w,F,g) and Mom (W,f,G)
☐ Dad (w,f,g) and Mom (w,F,G)

Figure 2. Example post-test question.

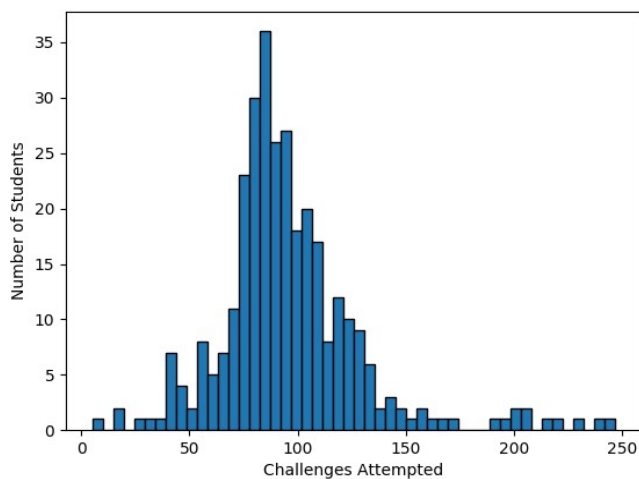


Figure 3. Histogram of students' gameplay trajectories.

evaluated in as few as one or as many as six items on the post-test survey since an assessment item can map to either one or two concepts. Item 28 is an open-ended question that can be answered in many unique ways, so we omit it from our competency score calculations. The mappings from each concept to individual survey questions can be found in Table 1.

Each of the test items is recorded in a binary format, with 1 if the student answered the item correctly and 0 if the item is answered incorrectly. To calculate the competency score for each concept, the total number of correctly answered items was divided by the total number of items for that concept, resulting in a score between 0 and 1. These scores serve as the target labels for our regression models.

5. METHODOLOGY

We evaluate two different approaches to the student competence modeling: single-task and multi-task. The single-task approach involves training an individual model for every concept, with each model only predicting a single competency score. This architecture allows for the model to focus exclusively on modeling trends and correlations between the students' gameplay features and a single competency and does not take into account any interrelationships between the gameplay and multiple concepts. The multi-task approach requires a single model trained to approximate all competency scores using a single 16-unit vector. This approach is advantageous as it is capable of modeling complex, non-linear relationships between the various concept-level competencies that exist within the gameplay data. Multi-task modeling has seen an increase in usage due to its reduced number of parameters to be estimated, as well as the computational time required to train a model for each dependent variable, compared to the single-task modeling technique. Multi-task models' capability to robustly model inherent relationships between multiple dependent variables using a shared input vector space makes this modeling technique ideal for stealth assessment frameworks, as well as circumstances where a large amount of training data may not be readily available [9].

We evaluate these two approaches using two different feature representations of the students' gameplay data: static and sequential representations. The static representation of the data involves producing a single feature vector representative of each student's overall interaction with the *Geniventure* learning environment, resulting in a single dataset of 316 total data samples. The sequential representation is used to model subsequences extracted from individual challenge-level interactions across each student's gameplay trajectory, retaining temporal information based on the order the challenges were completed. This sequence sub-sampling approach results in a single dataset of 29,977 total data samples.

5.1 Interaction Data

Gameplay interactions with the *Geniventure* game environment were recorded in a timestamped log file for each student. The trace data log is a raw event stream in JSON format which records fine-grained information about students' actions in the game, such as a navigated challenge, changed allele, submitted answer, and received hints from the system. The types of actions vary among challenges because of the differences in the challenge settings. To eliminate the influence of the differences in challenges, we defined 10 generic measurements across different challenges that describe contextual information about the challenge itself. The remaining features summarize students' performance and actions within an individual challenge. For each student, we generated his/her

gameplay trajectory across each individual challenge attempted. The length of the challenge level trajectories varied from 5 to 248 ($M = 95.86$, $SD = 33.63$). Each of the features forms the basis for the static and sequential data.

Table 1. Competency model concepts

Concept Number	Concept Description	Number of Questions
C1	Only one dominant allele is needed to produce the dominant trait.	3
C2	Two recessive alleles are needed to produce a recessive trait.	2
C3	Create or select parental gametes to create an individual offspring with a specific phenotype.	4
C4	Set parental genotypes to produce a specific pattern of offspring.	6
C5	Use patterns in the phenotypes of a group of offspring to predict the genotype of the parents.	5
C6	For some traits primarily influenced by a single gene, both alleles will have some effect, with neither being completely dominant.	2
C7	Breed with a recessive animal to determine an unknown genotype (testcross).	2
C8	Different versions of a gene correspond to (lead to the construction of) different versions of a specific protein.	2
C9	Proteins do work or have jobs to do in cells.	1
C10	Proteins are nanomachines; different proteins do different jobs.	1
C11	The function of a protein is determined by its shape.	1
C12	Different versions of a specific protein have different structures and may also have different functions.	1
C13	Some traits have multiple alleles, which can form a ranked series in terms of dominance.	2
C14	Genes on the X chromosome are referred to as X-linked. Males receive only one copy of the X chromosome and pass on their X only to their daughters.	1
C15	Working from the phenotype, determine possible genotypes for an organism.	2
C16	Use a genotype to predict the phenotype for an organism.	2

The features representing each challenge undertaken by a student are: (1) Pre-test score, (2) level of challenge, (3) mission number of challenge, (4) challenge number, (5) total time spent on challenge,

(6) number of movements made during challenge, (7) number of hints encountered during challenge, (8) number of correct movements made during challenge, (9) number of wrong movements made during challenge, and (10) student's completion status of challenge (0: incomplete, 1: complete with wrong answer, 2: complete with correct answer).

5.2 Static Competency Models

We evaluate five different regression models to determine their capabilities to predict students' competency levels for each concept. The features selected for the static competency models summarize the whole gameplay of each student across all challenges and levels. Using the challenge-level features noted above, the summative student-level features generated for the static models are (1) average time spent per challenge, (2) total time spent playing challenges, (3) fraction of challenges failed, (4) fraction of challenges succeeded, (5) fraction of challenges abandoned, (6) fraction of incorrect movements, (7) fraction of correct movements, (8) total hints received, (9) number of hints per level, (10) hint count per challenge, and (11) number of levels played.

We evaluate two variations of static modeling techniques: single-task and multi-task. Single-task models predict each target concept score as an independent regression problem. The data set and features used in each model are identical, but the target variable is a single competency score for each model. Multi-task models approximate all target variables in a single model. However, not all of the static, single-task models can effectively translate to a multi-task learning environment. Using single-task learning, we aim to discover the best model for each target variable independently while multi-task models perform better when there are underlying dependencies between the various competencies and a student's gameplay features.

5.2.1 Single-Task Models

We evaluate three single-task models. Elastic Net is a linear regression model that utilizes both L1 and L2 regularizations. The hyperparameter tuning of Elastic-Net was performed on the L1 and L2 regularization coefficients (alpha, L1 ratio). Gradient-Boosted Regression (GBR) is a decision tree-based modeling approach that builds an ensemble of weak predictors to approximate the target variable. The model is built in an iterative fashion where each subsequent stage improves on the model created in the previous stage. The hyperparameter tuning for the GBR model was based on fine-tuning the maximum depth of each tree in the model and the total number of estimators added to the model. We also evaluate a Random Forest regressor, another type of ensemble learning method using a 'forest' of decision trees. Each tree is randomly assigned insensitivity to different features in the training data (i.e., feature bagging). This approach allows for larger model ensembles while avoiding overfitting. The hyperparameter tuning for Random Forest was performed on the maximum depth of each tree and the total number of trees in the forest. While both Random Forest and Gradient Boosted Trees are decision tree-based ensemble learners, a notable difference between these two models is how the trees are added to the ensemble. Within Random Forest models, trees are added independently while in GBR models, trees are added incrementally to compensate for the shortcomings of the previous iteration of models. For the single-task approach, we use a single model for each competency score. We keep the regression model type consistent across all competencies and the hyperparameter values consistent across models.

5.2.2 Multi-Task Models

Due to constraints in models, only certain types of algorithms support multi-task modeling. In this work, two types of multi-task regression models are tested: the multi-task version of Elastic Net and the multi-task version of Random Forest. The multi-task version of Elastic Net adds the constraint that the selected features in the model are the same for all the tasks. The Random Forest regressor is one of the few models that does not require any special modification to support multi-task learning due to the trees in the regressor being built on different subsamples of the dataset.

Each of the static models was implemented using the scikit-learn library in Python. The data set (316 samples) is divided randomly into an 80/20 split with 20% serving as a held-out test set to evaluate the models. The 80% split is used for training the models, with five-fold cross-validation being applied to determine the best model. The training and test splits remain consistent across all investigated models and configurations (e.g., static vs. sequential, single-task vs. multi-task) to ensure a fair comparison between models, as well as the five-fold cross-validation splits. The final hyperparameter values as a result of the cross-validation on the training data are shown in Table 2.

Table 2. Static model hyperparameters

Regression Model	Task Type	Hyperparameters
Elastic Net	Single-Task	alpha = 0.05 L1 Ratio = 0.9
Gradient Boosted Regression	Single-Task	Max Tree Depth = 2 Number of Trees = 20
Random Forest	Single-Task	Max Tree Depth = 3 Number of Trees = 250
Elastic Net	Multi-Task	alpha = 0.2 L1 Ratio = 0.9
Random Forest Regression	Multi-Task	Max Tree Depth = 2, Number of Trees = 200

5.3 Sequential Competency Models

We explore four different types of deep learning-based models to model sequential representations of each student’s gameplay information across attempted challenges. Here, the motivation is to determine whether providing sequential context for each of the student’s problem-solving behaviors induces higher performance when modeling the competencies. To provide further sequential information to each of the models, we generate additional temporal features averaged across all challenges completed up to the current challenge attempted by the user: (1) average time per challenge, (2) average movements per challenge, (3) average correct movements per challenge, (4) average incorrect movements per challenge, (5) average hint count per challenge, (6) average unsubmitted challenges, (7) average failed challenges, and (8) average successful challenges.

We use these features in addition to the 10 static challenge-level features described in Section 5.1 to provide a total of 18 features to each of the sequential models. The models used for both the single-task and multi-task sequential models are variants of recurrent neural networks including Long Short-Term Memory recurrent neural networks (LSTMs) [11] and Gated Recurrent Units (GRUs) [6], due to their capability to model both single-task and multi-task data. LSTMs utilize a sequence of memory blocks that each contain

an input gate, forget gate, and an output gate. The forget gate determines whether the previous memory block’s gradient is retained or discarded, thus allowing the LSTM to model long-term dependencies across temporal sequences, while the input and output gates modulate the input and output vectors, respectively. GRUs are mechanisms that provide the same “forgetting” functionality as LSTMs but contain fewer hyperparameters, utilizing an update gate and a reset gate. This allows GRUs to be more computationally efficient and sometimes more effective on less training data than LSTMs.

In addition to standard LSTMs, we evaluate bidirectional LSTMs (Bi-LSTMs) [32] as well as LSTMs implementing a self-attention mechanism (SA-LSTMs) [41]. Bidirectional LSTMs are a variation of LSTMs that contain two input layers on opposing sides of the hidden layer, allowing the model to retain temporal information based on the past and the future of the input sequence, as opposed to only the past. A self-attention LSTM provides additional temporal context beyond contiguous feature vectors by utilizing a weighted sum of hidden representations of the entire sequence.

Adopting the same manner used in training the static competency models, each model is optimized using 5-fold cross validation, where the data splits are consistent across both static and sequential models to ensure a fair comparison, and then evaluated with the held-out test set. The hyperparameters are tuned using an iterative grid search, and each model was trained for 200 epochs. The subsequences used to train each sequential model were sampled across the challenges completed by each student using a sequence length of 10, and a sampling stride of 1. We use front padding in each sequence during the subsampling process to allow the models to fit during the beginning of each sequence. The concept-level prediction made for each student was calculated by taking the average competency prediction value across an entire sequence. The sequential data modeling pipeline was implemented using Python, and the deep learning models were implemented using the Keras library with the TensorFlow backend. The hyperparameter tuning was performed across the number of hidden units in each model’s hidden layer, as well as the dropout rate in the hidden layer [9]. The final hyperparameter values as a result of the cross-validation on the training data are shown in Table 3.

5.3.1 Single-Task Models

To evaluate the single-task sequential modeling approach, we train 16 different independent models, with each model approximating a single competency score based on the gameplay features described in Sections 5.1 and 5.3. Using the cross-validation performance on the training data, we selected the optimal configuration for each model type based on the highest performance in terms of the average R^2 value across all competency scores.

5.3.2 Multi-Task Models

Because of the architecture of the sequential deep learning models, each single-task model type is also able to perform as a multi-task model, with the only change occurring within the output layer, as the number of output units is expanded to contain an individual output node for each concept, instead of a single concept. This allows a single model to simultaneously infer student competencies across all concepts. Similar to the single-task models, the optimal model configurations were selected based on the average R^2 score across all concepts.

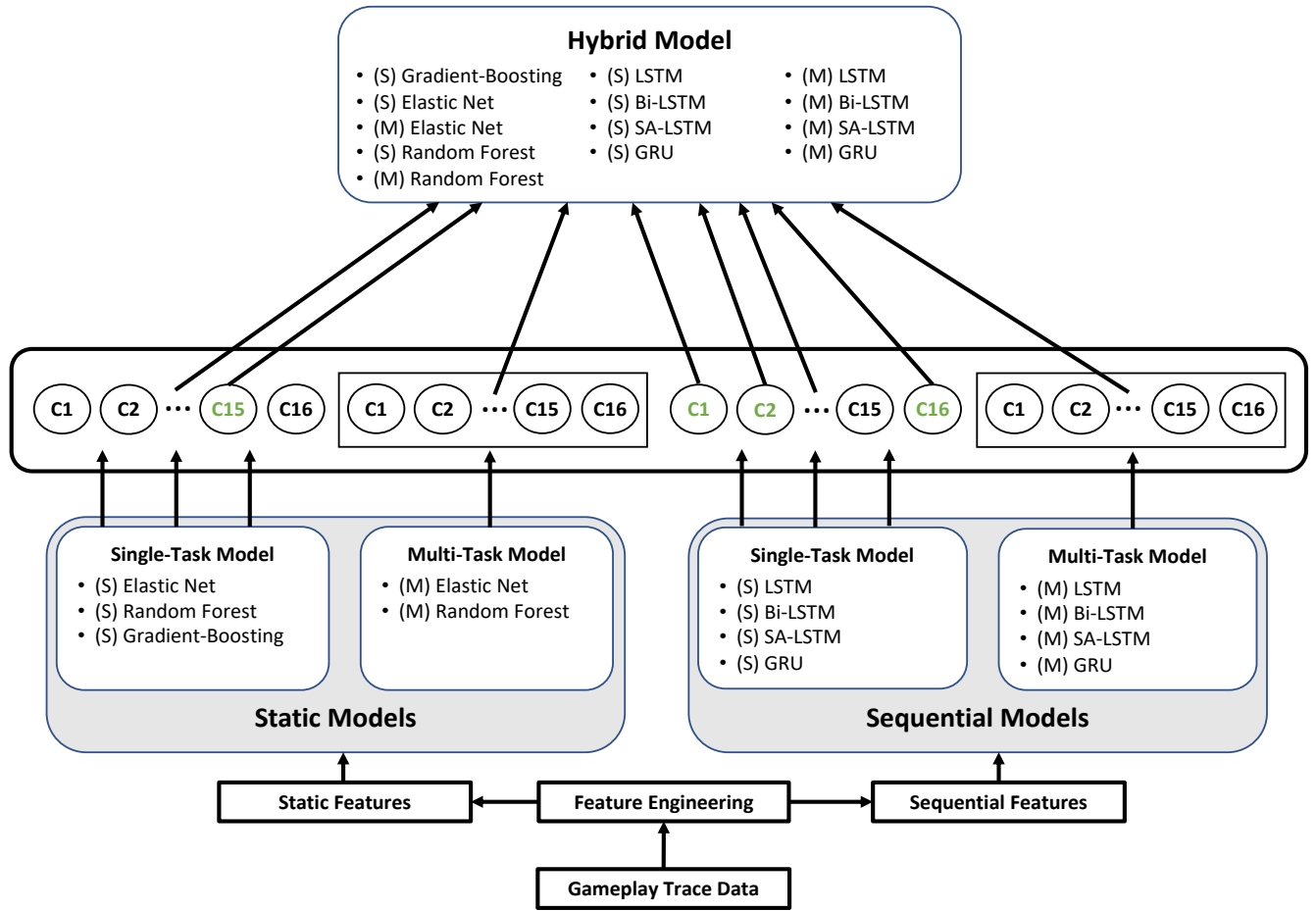


Figure 4. Hybrid, static, and sequential stealth assessment models.

We hypothesize that the optimal modeling techniques will vary due to the different complex characteristics that underlie each concept and the varying fitness of ECD models to these characteristics. Therefore, we propose the use of a hybrid framework that contains a combination of the various single-task and multi-task models that are both static and sequential. A visualization of the proposed hybrid stealth assessment framework comprised of the optimal models (highlighted in green text) is shown in Figure 4 above.

The rationale behind the use of a hybrid stealth assessment framework is that student competencies can vary widely with regards to each concept’s correlation to specific questions in post-test scoring methods, as well as each concept’s correlation to specific gameplay features or levels. By implementing both static and sequential variations of single-task and multi-task models, the long-term and short-term tendencies within each student’s gameplay is explored on a challenge and a student level. Additionally, the relationships between the individual competencies are modeled independently in the single-task approach, indicating whether certain concepts have no interweaving tendencies with other concepts within the gameplay. By utilizing a mixture of both single-task and multi-task models in this framework, multi-task models are only fit where underlying relationships exist between concepts, and concepts that have no underlying relationships with other concepts are optimally modeled by the single-task approach. The same concept applies to the sequential and static modeling: only concepts that have informative temporal trends across a student’s challenge-level gameplay data

are modeled by the sequential models. All other concepts are modeled by the static models utilizing only student-level data.

Table 3. Sequential model hyperparameters

Regression Model	Task Type	Hyperparameters
LSTM	Single-Task	Hidden units = 80 Dropout rate = 0.33
Bi-directional LSTM	Single-Task	Hidden units = 20 Dropout rate = 0.33
GRU	Single-Task	Hidden units = 80 Dropout rate = 0.5
Self-attention LSTM	Single-Task	Hidden units = 60 Dropout rate = 0.33
LSTM	Multi-Task	Hidden units = 100 Dropout rate = 0.5
Bi-directional LSTM	Multi-Task	Hidden units = 60 Dropout rate = 0.33
GRU	Multi-Task	Hidden units = 40 Dropout rate = 0.33
Self-attention LSTM	Multi-Task	Hidden units = 80 Dropout rate = 0.5

Table 4. R^2 value of single-task models based on held-out test set

Concept																
Model	C1	C2	C3	C4	C5	C6	C7	C8	C9	C11	C12	C13	C14	C15	C16	Mean
Elastic Net	0.210	0.169	0.176	0.300	0.348	0.232	0.145	0.185	0.073	0.138	0.048	0.098	0.052	0.163	0.094	0.162
GBR	0.214	0.301	0.156	0.363	0.365	0.119	0.107	0.190	0.045	0.111	0.034	0.110	0.081	0.263	0.100	0.171
RF	0.269	0.297	0.165	0.434	0.404	0.109	0.165	0.241	0.096	0.225	0.029	0.086	0.100	0.318	0.148	0.206
LSTM	0.314	0.383	0.149	0.398	0.346	0.155	0.141	0.157	0.054	0.091	0.034	0.013	0.028	0.302	0.185	0.183
Bi-LSTM	0.363	0.328	0.164	0.376	0.368	0.153	0.072	0.104	-0.029	0.075	0.009	-0.020	-0.094	0.273	0.262	0.160
SA-LSTM	0.315	0.351	0.135	0.356	0.314	0.148	0.107	0.123	0.009	0.070	0.050	0.038	-0.030	0.306	0.218	0.167
GRU	0.109	0.088	0.062	0.156	0.189	0.090	0.092	0.029	-0.158	0.017	0.022	0.004	-0.319	0.089	0.031	0.033

Table 5. R^2 value of multi-task models based on held-out test set

Concept																
Model	C1	C2	C3	C4	C5	C6	C7	C8	C9	C11	C12	C13	C14	C15	C16	Mean
RF	0.279	0.298	0.131	0.336	0.337	0.154	0.152	0.170	0.063	0.129	0.051	0.114	0.051	0.295	0.193	0.184
Elastic Net	0.211	0.171	0.182	0.307	0.350	0.239	0.159	0.181	0.077	0.136	0.057	0.099	0.049	0.172	0.080	0.165
LSTM	0.291	0.270	0.144	0.362	0.346	0.147	0.179	0.174	0.075	0.131	0.058	0.024	0.029	0.259	0.130	0.175
Bi-LSTM	0.313	0.273	0.157	0.371	0.356	0.142	0.176	0.166	0.066	0.123	0.058	0.021	0.014	0.260	0.144	0.176
SA-LSTM	0.320	0.302	0.176	0.361	0.352	0.173	0.201	0.133	0.023	0.110	0.017	0.034	-0.055	0.313	0.255	0.181
GRU	0.309	0.241	0.152	0.352	0.350	0.156	0.199	0.169	0.061	0.127	0.050	0.048	0.036	0.275	0.040	0.171

Table 6. Highest R^2 values of optimal hybrid competency models

Concept																
Model	C1	C2	C3	C4	C5	C6	C7	C8	C9	C11	C12	C13	C14	C15	C16	Mean
Hybrid	0.363	0.383	0.182	0.434	0.404	0.239	0.201	0.241	0.096	0.225	0.058	0.114	0.100	0.318	0.262	0.241

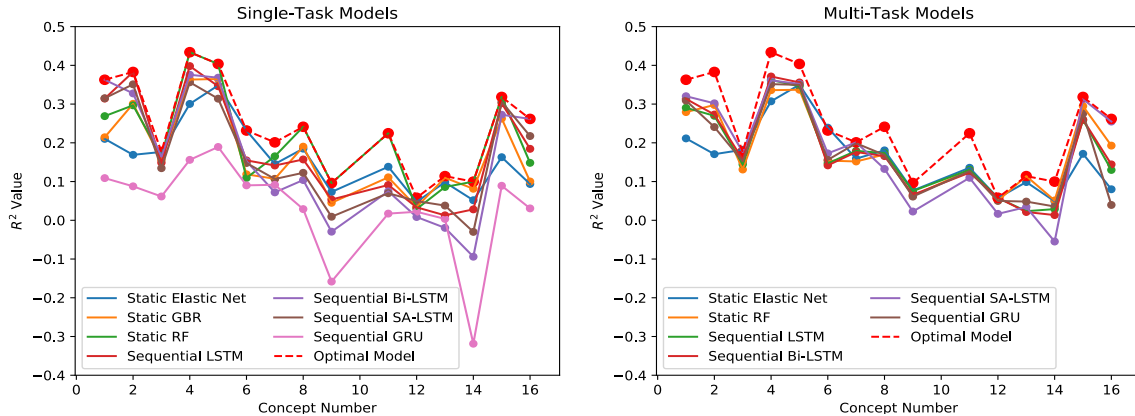


Figure 5. Performance of single-task and multi-task models compared to hybrid model performance.

6. RESULTS AND DISCUSSION

We report the results of the single-task models (Table 4) and the multi-task models (Table 5) for each concept in terms of R^2 . The highest R^2 value produced for each individual concept is presented in Table 6, as this represents the performance of our proposed hybrid framework across all concepts. Figure 5 shows the performance of single-task and multi-task models compared to the performance of the optimal hybrid model (Left: comparison to single-task models, Right: comparison to multi-task models). The results are obtained based on each model's performance on the held-out test set after being trained on the entirety of the training set. As noted above, the cross-validation splits applied to the training set were performed on a student level to prevent data

leakage and were consistently applied to the set of machine learning techniques for a fair comparison across different models. For this work, Concept 10 was omitted because every student that took the post-test survey answered the question correctly, resulting in a dependent variable with zero variance, thus having no impact on the evaluation of our respective models.

The best performing model in terms of average R^2 value across all concepts was the single-task Random Forest; however, it was only the optimal model for 7 out of the 15 total concepts. The single-task bidirectional long short-term memory network performed the highest for two concepts, as well as the multi-task Elastic Net. The single-task LSTM and the multi-task Random Forest (RF), Bi-LSTM, and self-attention LSTM were optimal models for one concept each. The Gradient Boosted Regression (GBR), single-task

Gated Recurrent Unit, and single-task SA-LSTM performed relatively poorly and were not the highest performing competency models for any of the concepts. The multi-task models were the most effective approach for five of the 15 concepts, while single-task models were most effective for the other 10 concepts.

Across both single-task and multi-task models, the GRU was the lowest-performing model, achieving an average R^2 value of 0.102 across all concepts. The results that variants of LSTMs (e.g., standard LSTMs, SA-LSTMs, Bi-LSTMs) achieved the highest R^2 score in predicting student competencies on at least one concept demonstrate that there exist complex, sequential patterns within the students' gameplay data, which were effectively modeled by the LSTMs' three gating units, but not by the two gating units enabled in GRUs. The SA-LSTMs, Bi-LSTMs, and standard LSTM models all returned relatively equal performances across the single-task and multi-task data, with average R^2 values of 0.174, 0.168, and 0.179, respectively. It appears that although the Bi-LSTM and SA-LSTM capture various extra temporal contextual patterns not inherently captured by the standard LSTM, this information is not globally beneficial to all the competency models, explaining why neither model outperforms the standard LSTM on average. However, this result might also be attributed to the fact that the sequential data was only generated from 316 students, which may not be enough information for any of the more complex, sequential models utilizing a higher number of trainable parameters, to truly detect informative underlying temporal patterns.

Selecting the single-task RF as the model for all concepts based on its average performance across all the concepts results in a mean R^2 value of 0.206. However, as illustrated in Table 6, by using our proposed hybrid system approach and selecting the optimal model for each individual concept, we can obtain a performance of 0.241, which is a 17.0% improvement compared to a homogenous framework typically used within stealth assessment. Our observation that the use of multiple models in the hybrid stealth assessment framework would induce higher performance than using a single model can be explained by the fact that static, sequential, single-task, and multi-task models were all selected as an optimal model at least once.

Additionally, it should be noted that when considering only the concepts that mapped to multiple questions (i.e. 1-8, 13, 15-16), the deep-learning based sequential models produced a higher and more consistent performance (0.224 for single-task, 0.222 for multi-task) than the static models (0.214 for single-task, 0.210 for multi-task) on average. The multi-task SA-LSTM and the single-task RF both achieved the optimal performance across the multi-question concepts, with an average R^2 value of 0.239. Random Forest may also perform relatively well as a competence model because it uses an ensemble approach, making it more robust against overfitting.

One correlation that was noted is that the single-task models were the best technique for 75% (3 out of 4) of the concepts that had only one corresponding question in the post-test. This can potentially be attributed to the fact that each of the single-task models only models a single concept, without taking into account any of the linear and non-linear relationships that might exist between the gameplay features and the different competencies for a single student. Concepts that correspond to only a single question possibly contain a less complex relationship between the competency scores and the gameplay features, meaning that a single-task model is sufficient for that modeling task without simultaneously modeling any context related to competencies for other concepts, which can have a detrimental impact to the predictive tasks. In addition, each of the three optimal models for the single-question concepts were trained

using static feature representations, suggesting that the student-level features were the most informative to our model, and the temporal information did not yield greater predictive performance for the student competency models.

However, we also observe that the single-task models were also frequently the highest-performing models for the multi-question concepts. Seven out of the 11 concepts that were represented by multiple post-test questions were optimally modeled using a single-task model, either using static or sequential representations. Out of these 7 highest-performing models, 4 of them used static input representations. In a similar manner to the single-task models mentioned previously, this implies that student-level features were informative for a subset of the multi-question concepts, while the temporal context provided within the sequence modeling tasks was still beneficial to predicting students' individual competencies for the three other concepts.

Overall, the majority of optimal classifiers across the single-question and multi-question concepts were single-task, static representations, as these account for 7 out of the 15 total concepts we evaluated. We then analyze the remaining models to investigate if there are any correlations between the concepts and the optimal models. The competency models for Concepts 1 and 2 were both modeled using sequential single-task models, two concepts that correspond to five combined post-test questions. Concept 1 deals with generating dominant traits using alleles, while Concept 2 deals with a similar task generating recessive traits using alleles. The similarity in these two concepts may be a possible reason that the highest predictive performance was achieved by the same modeling approach. The highest R^2 values (0.434 and 0.404) occurred in Concepts 4 and 5, which are the two concepts that correspond to 6 and 5 post-test questions, respectively. The correlation between the higher performance in these two RF-based competency models can be explained by the fact that ensemble models leveraging more single-task models contribute to improvement of the average predictive performance, which prevents a model that produces a less accurate prediction from heavily impacting the overall representative performance.

The relative scores between concepts are highly correlated across modeling methods. In other words, the concepts that had a high R^2 score for one model also had a high R^2 score for most of the other models. As shown in Figure 5, Concepts 4 and 5 have the highest R^2 value irrespective of the modeling method, and Concept 14 is on the lower end of R^2 values. This could be because of how well a gameplay feature predicts a concept is dependent on the type of concept. In other words, some concepts are harder to model irrespective of the modeling approach used for the model.

Interestingly, concepts that contained only a single question (i.e. 9, 11-12, 14) produced noticeably low R^2 values. These single-question concepts produced an average R^2 value of 0.046 across all the models. Because there was only a single question associated with the concept, each competency score was entirely dependent on students' single response to the question, which could result in a reliability issue in the competency scores due to students' behaviors related to guess and slip as well as a higher variance in the scores, together possibly attributing to these low R^2 scores.

A chart of the average student score for each concept based on their post-test performance is shown in Figure 6 below, distinguishing between single-question and multi-question concepts. The average student performance on multi-question concepts was markedly higher than for single-question concepts, with students achieving scores of 0.672 and 0.507, respectively. It was noted that the

average students' scores on the questions mapped to a single concept was remarkably low compared to an overall student score of 0.661 across all questions. This factor may also have impacted the predictive performance of the competency models compared to competency models that encounter fairly consistent or accurate student answers to post-test questions, as the questions corresponding to lower student scores introduce higher variance into the resulting competency scores used to train the models.

Finally, we investigate the impact that overlapping concepts may have on the performance of the classifiers. A concept is considered to be "overlapping" if it shares a correlated question with one or more different concepts. Out of the 16 concepts, 6 concepts were found to be "completely" overlapping; that is, every question associated with that concept was also associated with another concept. One concept was "partially" overlapping, indicating that only a portion of the associated questions were also mapped to another concept. The remaining concepts were the only ones that corresponded with their own associated question or group of questions. Student scores were significantly higher for overlapping concepts as opposed to non-overlapping questions, achieving average scores of 0.718 and 0.550, respectively. This trend is also present in our competency models, as the optimal models in our hybrid framework yield R^2 values of 0.339 (overlapping) and 0.157 (non-overlapping) on average. Surprisingly, the optimal models for the overlapping concepts were primarily single-task models, with the exception of one model. This indicates that multi-task modeling across all the concepts including both relevant and irrelevant concepts is actually detrimental in terms of achieving higher predictive performance. Thus, a promising future direction is to investigate multi-task learning performance by grouping relevant concepts and separately modeling related concepts only.

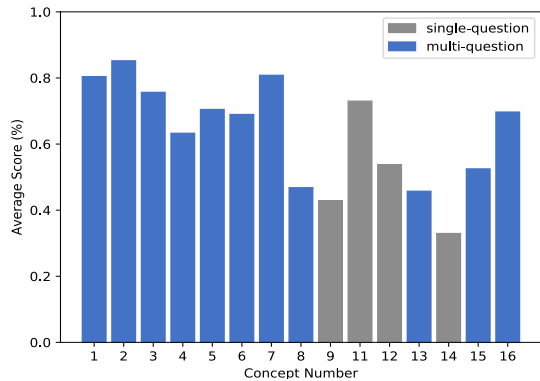


Figure 6. Student post-test performance on concepts associated with single and multiple questions.

In this particular application domain (genetics), concepts C1 and C2 are foundational to eight other concepts, as they describe a common pattern of gene variant behavior in inheritance of traits. Some concepts are related variously to other concepts, e.g., C5 requires deductive reasoning based on C1, C2, and C15, while it also serves as a prerequisite for C7, which allows determining gene variants for ambiguous traits. Alternatively, concepts C9-C12 focus primarily on molecular genetic inheritance and are not as tightly related to other concepts. This example of varying connections within genetics-related concepts illustrates the broader application of our hybrid model and why it demonstrates promise for other domains.

7. CONCLUSION

Stealth assessment holds considerable potential for game-based learning. Recent work exploring stealth assessment has typically

employed a single machine learning technique to devise competency and evidence models. This approach operates under the assumption that each student competency can be optimally modeled by the same learning algorithm that yields the highest predictive performance on average. However, this may not always be the case, as student competencies often have varying interleaving relationships with each other or even underlying complexities within itself.

In this work, we demonstrate the effectiveness of a hybrid stealth assessment framework consisting of a combination of single-task and multi-task models, using static and sequential features to represent student gameplay data. We evaluate our stealth assessment framework using a game-based learning environment and predict student competencies as measured by a post-test. Results indicate that a heterogeneous approach to stealth assessment modeling techniques induces higher results across all concepts when compared to the single-model baseline evaluations. Selecting a single competency model for all concepts based on its average performance across all the concepts is a common practice in stealth assessment frameworks. However, the proposed hybrid system using the optimal model for each individual concept returns a performance that is substantially higher than a homogeneous framework. In addition to static, single-task modeling, the sequential, multi-task modeling approach can adapt to multiple concepts by effectively capturing sequential context underlying individual students' gameplay behaviors, as well as simultaneously modeling various competencies that were manifested throughout the gameplay sessions. The use of all of the aforementioned modeling techniques provides a multi-dimensional approach that has been demonstrated to be a step forward in improving stealth assessment techniques.

There are a number of future directions that can be investigated to further improve the performance of the hybrid stealth assessment framework. Multi-task learning becomes increasingly difficult as the number of tasks increases and training deep sequential models for 16 tasks using only 316 data samples is likely a limiting factor in the multi-task models' performances. To gain further insight into the use of multi-task learning as a competency modeling technique, the hybrid stealth assessment framework presented in this work should be evaluated on comparatively larger datasets. This also enables the evaluation of the hybrid framework's ability to adequately translate to other student populations. Alternatively, different ways to reduce the number of tasks can be investigated. Due to the hierarchical, interweaving relationships within both individual concepts and between concepts and various questions, it will be worthwhile to investigate other sophisticated hierarchical modeling methods such as Bayesian hierarchical modeling or clustering methods, as well as refine the post-test questions and the mapping to the concepts to more reliably assess students' competency for each concept. Additionally, the feature engineering process performed for both static and sequential models can evolve significantly, possibly inducing higher performance from the competency models. Finally, it will be instructive to investigate the generalizability of this framework across different learning environments, contexts, and student populations.

8. ACKNOWLEDGEMENTS

The authors would like to thank Robert Taylor for his assistance in facilitating this research. This research was supported by the National Science Foundation under Grant DRL-1503311. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

9. REFERENCES

- [1] Akram, B., Min, W., Wiebe, E., Mott, B., Boyer, K. E., and Lester, J. 2018. Improving stealth assessment in game-based learning with LSTM-based analytics. In *Proceedings of the 11th International Conference on Educational Data Mining*. International Educational Data Mining Society, 208-218.
- [2] Asbell-Clarke, J., Rowe, E., Sylvan, E., and Baker, R. 2013. Working through impulse: assessment of emergent learning in a physics game. *Games+ Learning+ Society*. 9.
- [3] Bosch, N., Chen, H., D'Mello, S., Baker, R. and Shute, V. 2015. Accuracy vs. availability heuristic in multimodal affect detection in the wild. In *Proceedings of the 2015 International Conference on Multimodal Interaction*. 267-274.
- [4] Buffum, P. S., Frankosky, M., Boyer, K. E., Wiebe, E. N., Mott, B. W., and Lester, J. C. 2016. Collaboration and gender equity in game-based learning for middle school computer science. *Computing in Science and Engineering*. 18, 2, 18-28.
- [5] Chaudhry, R., Singh, H., Dogga, P., and Saini, S. K. 2018. Modeling hint-taking behavior and knowledge state of students with multi-task learning. In *Proceedings of the International Conference on Educational Data Mining*. International Educational Data Mining Society.
- [6] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- [7] Clark, D. B., Tanner-Smith, E. E., and Killingsworth, S. S. 2016. Digital games, design, and learning: A systematic review and meta-analysis. *Review of Educational Research*. 86, 1, 79-122.
- [8] Falakmasir, M.H., Gonzalez-Brenes, J.P., Gordon, G.J. and DiCerbo, K.E. 2016, April. A data-driven approach for inferring student proficiency from game activity logs. In *Proceedings of the Third ACM Conference on Learning@Scale*. 341-349.
- [9] Geden, M., Emerson, A., Rowe, J., Azevedo, R., and Lester, J. 2020 (in press). Predictive student modeling in educational games with multi-task learning. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*.
- [10] Hamari, J., Shernoff, D. J., Rowe, E., Coller, B., Asbell-Clarke, J., and Edwards, T. 2016. Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Computers in Human Behavior*. 54, 170-179.
- [11] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*. 9, 8, 1735-1780.
- [12] Karumbaiah, S., Baker, R. S., and Shute, V. 2018. Predicting quitting in students playing a learning game. In *Proceedings of the 11th International Conference on Educational Data Mining*. 167-176.
- [13] Kiili, K., Devlin, K., Perttula, T., Tuomi, P., and Lindstedt, A. 2015. Using video games to combine learning and assessment in mathematics education. *International Journal of Serious Games*. 2, 4, 37-55.
- [14] Kim, Y.J., Almond, R.G. and Shute, V.J. 2016. Applying evidence-centered design for the development of game-based assessments in physics playground. *International Journal of Testing*. 16, 2, 142-163.
- [15] Lester, J.C., Ha, E.Y., Lee, S.Y., Mott, B.W., Rowe, J.P. and Sabourin, J.L. 2013. Serious games get smart: Intelligent game-based learning environments. *AI Magazine*. 34, 4, 31-45.
- [16] Liu, Z., Zhi, R., Hicks, A., and Barnes, T. 2017. Understanding problem solving behavior of 6-8 graders in a debugging game. *Computer Science Education*. 27, 1, 1-29.
- [17] Ma, Y., Cui, C., Yu, J., Guo, J., Yang, G., and Yin, Y. 2019. Multi-task MIML learning for pre-course student performance prediction. *Frontiers of Computer Science*. 14, 5, 145313.
- [18] McElroy-Brown, K. and Reichsman, F. 2019. Genetics with dragons: Using an online learning environment to help students achieve a multilevel understanding of genetics. Retrieved from <http://concord.org>.
- [19] Mills, C., D'Mello, S., Lehman, B., Bosch, N., Strain, A., and Graesser, A. 2013. What makes learning fun? exploring the influence of choice and difficulty on mind wandering and engagement during learning. In *Proceedings of the International Conference on Artificial Intelligence in Education*. Springer, 71-80.
- [20] Min, W., Frankosky, M., Mott, B.W., Rowe, J., Smith, P.A.M., Wiebe, E., Boyer, K., and Lester, J. 2019. DeepStealth: game-based learning stealth assessment with deep neural networks. *IEEE Transactions on Learning Technologies*.
- [21] Min, W., Frankosky, M. H., Mott, B. W., Wiebe, E. N., Boyer, K. E., and Lester, J. C. 2017. Inducing stealth assessors from game interaction data. In *Proceedings of the 9th International Conference on Artificial Intelligence in Education*. Springer, 212-223.
- [22] Min, W., Frankosky, M.H., Mott, B.W., Rowe, J.P., Wiebe, E., Boyer, K.E. and Lester, J.C. 2015. DeepStealth: leveraging deep learning models for stealth assessment in game-based learning environments. In *Proceedings of the International Conference on Artificial Intelligence in Education*. Springer, Cham, 277-286.
- [23] Min, W., Mott, B. W., Rowe, J. P., Liu, B., and Lester, J. C. 2016. Player goal recognition in open-world digital games with long short-term memory networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 2590-2596.
- [24] Mislevy, R., Steinberg, L., and Almond R. 2003. Focus article: on the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives*. 1, 1, 3-62.
- [25] Owen, V. E., Roy, M. H., Thai, K. P., Burnett, V., Jacobs, D., Keylor, E., and Baker, R. S. 2019. Detecting wheel-spinning and productive persistence in educational games. International Educational Data Mining Society.
- [26] Puntambekar, S. and Hubscher, R. 2005. Tools for scaffolding students in a complex learning environment: What have we gained and what have we missed?. *Educational Psychologist*. 40, 1, 1-12.
- [27] Qu, S., Li, K., Wu, B., Zhang, X., and Zhu, K. 2019. Predicting student performance and deficiency in mastering knowledge points in MOOCs using multi-task learning. *Entropy*. 21, 12, 1216.

- [28] Roschelle, J., Dimitriadis, Y. and Hoppe, U. 2013. Classroom orchestration: synthesis. *Computers & Education*. 69, 523-526.
- [29] Rosenheck, L., Lin, C., Klopfer, E., and Cheng., M. 2017. Analyzing gameplay data to inform feedback loops in the radix endeavor. *Computers & Education*. 111, 60–73.
- [30] Rowe, J. P., Shores, L. R., Mott, B. W., and Lester, J. C. 2011. Integrating learning, problem solving, and engagement in narrative-centered learning environments. *International Journal of Artificial Intelligence in Education*. 21, 1-2, 115-133.
- [31] Sawyer, R., Smith, A., Rowe, J., Azevedo, R. and Lester, J. 2017. Enhancing student models in game-based learning with facial expression recognition. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. 192-201.
- [32] Schuster, M. and Paliwal, K. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*. 45, 2673 - 2681.
- [33] Shute, V. J. and Ke, F. 2012. Games, learning, and assessment. *Assessment in Game-Based Learning*. Springer, 43-58.
- [34] Shute, V. J. 2011. Stealth assessment in computer-based games to support learning. *Computer Games and Instruction*. 55, 2, 503-524.
- [35] Shute, V., Ventura, M., Zapata-Rivera, D., and Bauer, M. 2009. Melding the power of serious games and embedded assessment to monitor and foster learning flow and grow. *Serious Games: Mechanisms and Effects*. 2, 295–321.
- [36] Shute, V. and Ventura, M. 2013. *Measuring and Supporting Learning in Games: Stealth Assessment*. The MIT Press, Cambridge, MA.
- [37] Shute, V.J. and Moore, G.R. 2017. Consistency and validity in game-based stealth assessment. *Technology enhanced innovative assessment: Development, Modeling, and Scoring from an Interdisciplinary Perspective*. 296.
- [38] States, N. L. 2013. Next Generation Science Standards. Washington.
- [39] Tabak, I. 2004. Synergy: A complement to emerging patterns of distributed scaffolding. *The Journal of the Learning Sciences*. 13, 3, 305-335.
- [40] Tokac, U., Novak, E., and Thompson, C. G. 2019. Effects of game-based learning on students' mathematics achievement: A meta-analysis. *Journal of Computer Assisted Learning*. 35, 3, 407-420.
- [41] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998-6008.
- [42] Wang, P., Rowe, J.P., Min, W., Mott, B.W. and Lester, J.C. 2017. Interactive narrative personalization with deep reinforcement learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 3852-3858.
- [43] Wouters, P., Van Nimwegen, C., Van Oostendorp, H., and Van Der Spek, E. D. 1993. A meta-analysis of the cognitive and motivational effects of serious games. *Journal of Educational Psychology*. 105, 2, 249-267.
- [44] Wouters, P. and Van Oostendorp, H. 2013. A meta-analytic review of the role of instructional support in game-based learning. *Computers & Education*. 60, 1, 412-425.

Harbingers of Collaboration? The Role of Early-class Behaviors in Predicting Collaborative Problem Solving

Paul Hur
University of Illinois at
Urbana-Champaign
khur4@illinois.edu

Nigel Bosch
University of Illinois at
Urbana-Champaign
pnb@illinois.edu

Luc Paquette
University of Illinois at
Urbana-Champaign
lpaq@illinois.edu

Emma Mercier
University of Illinois at
Urbana-Champaign
mercier@illinois.edu

ABSTRACT

Collaborative problem solving behaviors are difficult to identify and foster due to their amorphous and dynamic nature. In this paper, we investigate the value of considering early class period behaviors, based on small group development theory, for building predictive machine learning models of collaborative behaviors during problem solving. Over 12 weeks, 20 small groups of undergraduate students solved problems facilitated by a digital joint problem space tool on tablet computers, in the 50-minute discussion component of an engineering course. We annotated 16,270 video clips of groups for collaborative behaviors including task relatedness, talk content, peer interaction, teaching assistant interaction, and tablet usage. We engineered two subsets of features from tablet log file data: onset features (early collaborative problem solving behavior characteristics calculated from the first ten minutes of the class) and concurrent features (more general collaborative behaviors from the whole class period). We compared accuracy between the onset, concurrent, and onset + concurrent features in machine learning models. Results exhibited a U-shaped pattern of accuracy over class time, and showed that onset features alone could not be used to effectively model groups' collaborative behaviors over the entire class time. Furthermore, analysis did not show support for significant gain in accuracy when onset features were combined with concurrent features. Finally, we discuss implications for studying collaborative learning and development of software to facilitate collaboration.

Keywords

Collaborative Problem Solving, Computer-Supported Collaborative Learning, Predicting Collaboration, Small Group Development

Paul Hur, Nigel Bosch, Luc Paquette and Emma Mercier "Harbingers of Collaboration? The Role of Early-Class Behaviors in Predicting Collaborative Problem Solving" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 104 - 114

1. INTRODUCTION

Collaborative problem solving consists of the communication and coordination of shared effort between team members toward a common desired goal [19, 23, 26]. Though it has been identified as a critical skill for students in the classroom [11, 34, 25], it is difficult to identify effective behaviors and nurture them, since the nature of collaboration and teams can be amorphous [48] and dynamic [43]. Education and learning sciences researchers have advocated for qualitative coding of video data as a means to understand the complexities of learning behaviors [24], and have applied these methods to study collaborative behaviors and the development of collaborative practices in courses [35]. Computers can further support collaborative learning research through collaborative learning software, collaborative games, and digital joint problem spaces—"a socially-negotiated set of knowledge elements, such as goals, problem state descriptions and problem solving actions" [44]—the resulting log data of which have been widely used with machine learning and data mining approaches to uncover hidden patterns of collaborative behaviors [31, 1, 37, 12]. Recent technological advances have also given way to multimodal approaches, using eye-gaze tracking, bodily motion, and physiological data to identify collaborative states [28, 40].

Despite such diverse approaches to detect and identify collaborative behaviors in learning contexts, the evolution of collaborative practices in student groups has not been closely investigated. Understanding the evolution of collaboration and its impact on methods for measuring collaboration is crucial, however. What constitutes collaborative behaviors may change throughout a learning session [10], and thus measurement may need to be adapted as well. In this paper we focus on the relationship between measurement and behavioral changes over time within classroom sessions. In particular, we leverage organizational theory about the sequential nature of small group development to inform research on how to measure and predict collaboration via machine learning in the presence of inevitable shifts in behaviors throughout collaboration stages.

The rest of the paper is organized as follows: we first discuss the small group development theories on structured, sequential group development which motivated our work, then relate

them to collaborative problem solving in the classroom to define our research questions and respective hypotheses. We then introduce the context of our study, including the collaboration tool, behavior coding, data processing, and model building. Next, we present our findings and close with an interpretation of our results and note limitations and future work.

1.1 Small Group Development

Research in organization and management fields on understanding how collaborative behaviors contribute to small group dynamics and development goes back several decades. Perhaps most notably, Tuckman's 1965 meta-analysis of therapy and human relations training groups presented the *forming-storming-norming-performing* (and later a fifth stage, *adjourning* [50]) model [49]. The model outlined the existence of a sequential, stage-based trajectory of small group collaboration, in which a group must fulfill one stage before advancing to the next. Tuckman's five model stages were described as (1) orientation to task (forming), (2) emotional response to task demands (storming), (3) open exchange of relevant interpretations (norming), (4) emergence of solutions (performing), and (5) separation (adjourning). This has led to decades of efforts to better understand the stages in various settings, including management [36], education [51], and medical training [47].

Tuckman's 5-stage structure of group development was further supported by Cassidy's 36-book meta-framework study, which aimed to clarify group development for practical use by examining group development in therapy, education, and management settings [17]. Though some scholars have presented theoretical models with more or fewer stages to group development [46, 21, 54], others have supported the five-stage model with differently termed, but analogous stages to Tuckman's model [16, 22, 8].

In nearly all proposed theoretical models of small group development, the first stage is defined as the task orientation stage [49, 16, 22]. During this stage, group members contextualize the task within the given parameters and communicate regarding the manner in which it will be accomplished [49]. While "ground rules" are set during this stage, communication about task orientation continues on some level throughout the collaboration process. Moreover, in problem solving, communication with references to others' ideas rather than independent solution paths has been identified as an important marker of shared task alignment, or "establishment of a collaborative orientation toward problem solving" [4]. In this study, we briefly analyze transitions across the stages of small group development during problem solving in classrooms. However, we focus much more closely on the first stage, orientation to task, since it has been shown to have a significant positive effect on achievement [45]. The first stage characterizes cooperative orientation and the motivation to collaborate, which has a strong relation to the quality of collaboration [13].

1.2 Contributions and Novelty

This paper considers the role of early group behaviors in collaborative problem solving. We investigated whether explicitly incorporating early group behaviors as features improves machine learning predictions of collaboration and

analyze how model accuracy evolves across time and stages of collaboration.

We used qualitative coding of collaborative behaviors on video data to measure collaboration. We then predicted those behaviors from features extracted from the action log files of a digital collaboration tool (run on tablet computers) used by undergraduate students in an introductory mechanical engineering course at a large Midwestern U.S. research university. We created various feature subsets and built corresponding machine learning models to evaluate the predictive accuracy of early group behaviors versus behaviors from later on in class periods. Assuming the presence of sequential, evolving collaborative behaviors in small groups, and the importance of early collaborative behaviors, machine learning models created from considering class behaviors as a whole may potentially be improved by accounting for early behaviors. For example, a group of students who fail to form a successful collaborative dynamic early on may struggle throughout class, whereas a group of students who exhibit high collaboration early on may be more effective in later stages. Consequently, we analyze whether a model built on features from class behaviors as a whole would have variable performance for collaborative behaviors predictions over the different segments of the class period, which align with the different stages of group development.

We aim to understand how effective collaborative behaviors, relating to orientation to task, during earlier stages may influence a group's collaborative behaviors in the future. As such, we also compare the performance a model solely built from such earlier features with one built from features of behaviors from all current and past in-class behaviors, not just early-stage behaviors.

We approach the aim of this paper by formulating and addressing several research questions:

RQ1 How does the predictive accuracy of collaborative behaviors vary across different periods of a 50-minute class?

Hypothesis: We expect stages of collaboration that are dominated by tablet computer interaction behaviors (e.g., reading, drawing) will be more successfully predicted than those dominated by discussion, and that the changing base rates of collaborative behaviors over time will influence classification accuracy [29].

RQ2 Can early class collaborative behaviors alone be used to effectively model and predict collaborative behaviors of the entire class period?

Hypothesis: We expect early class behaviors to predict the quality of collaboration later in class if and only if groups' collaboration quality remains static or consistently mirrors early collaboration.

RQ3 Are collaborative behavior prediction models improved through emphasizing early class collaborative behavior features?

Hypothesis: We expect prediction models will be more accurate later in class periods if early class behaviors capture

groups that are consistently collaborative or consistently not collaborative.

2. RELATED WORK

In this study, we utilized video coding methods along with machine learning approaches for analyzing action log data to study temporality. Work in computer-supported collaborative learning (CSCL) has highlighted the importance of considering temporality in collaboration, and Reimann has argued that “the main object of analysis in CSCL is a process—something that unfolds over time” [42]. For example, Mercier et al. examined through video coding and counting how the development of collaborative practices in engineering courses evolve over four weeks, and saw that patterns of interactions, such as conversation and workflow, change over time [35]. Others highlighted the value of utilizing more complex quantitative methods over video coding methods to consider temporality in analyzing problem-solving processes in computer-supported collaboration settings, as it can reveal aspects of group interactions that coding methods cannot reveal [32].

Collaborative learning may also be effectively analyzed via the action logs, discourse data, and gameplay data of digital tools and serious games, which are able to provide fine-grain recollections of the learner’s interactions with the respective software. Educational data mining researchers have applied supervised [41] and unsupervised [14, 31] machine learning techniques to better understand collaboration and to inform the design of interventions to support collaborative learning through such means as software prompts [31] and content creation suggestions [52]. Additionally, Paquette et al. have highlighted the need to support students during collaborative learning by considering the role of the instructor in facilitating student collaboration [38]. As such, instructor dashboards have been explored as ways for instructors to more easily gauge and analyze student collaboration across multiple groups [3, 33]. A central aim of our study has been to inform better instructor interventions for facilitating collaboration through insights gained from analysis of action log data.

3. METHODS

This study utilizes data collected from a design-based implementation research project which aims to better facilitate collaboration in engineering problem solving through the analysis of video and interactions from engineering classes. The project team has developed a student-facing tool that facilitates student group collaboration through a synchronized-per-group shared digital environment (Figure 1) on tablet computers, which group members can use to create and display their work. During use, we collected two types of data: student interactions on the tool stored in log files—detailing actions taken by individual students such as writing, drawing, or editing—and video data from cameras set up around the classroom. One of the key goals of the tool is to scale to large classrooms where cameras are unlikely to be consistently available; thus, we utilize video data to collect ground truth labels, but rely only on logged tablet actions for collaboration prediction.

Data in this study came from the use of the tool in Fall 2017 during the discussion component of an undergraduate intro-

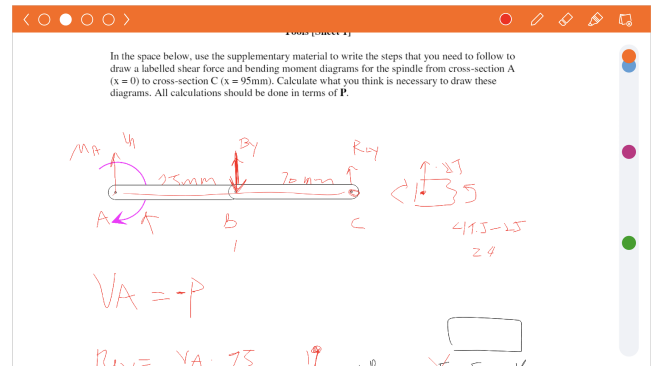


Figure 1: One example of the result of collaborative problem solving through the tool’s shared digital environment. The interface allows students choices of different colors and tools to write, draw, and create figures.

ductory mechanical engineering course at a large Midwestern U.S. research university. The research team worked closely with faculty and teaching assistants (TAs) to design tasks suitable for collaboration and in line with the intended learning outcomes from the class. The tasks were independent from week-to-week and did not build on one another, and the students were not graded on completion by the end of each class period. The tasks were represented in the tablet tool as worksheets with variable number of pages, which included problem descriptions and space to work out solutions. Data were collected across 12 weeks of class from 20 groups of approximately 4 students (group sizes varied from week to week based on attendance).

While students interacted on tablets using the interface shown in Figure 1, TAs present in the classroom viewed student progress on their own tablets (Figure 2). The TA tablets showed students’ editing positions in the worksheets, and allowed TAs to join any group as a non-interactive participant to see students’ work in detail. Our current work seeks to augment the TA-facing tool via predictions of various markers of collaboration quality made by machine learning models. This feature enables TAs, who may lack extensive training in assessing and promoting collaboration, to identify groups that are not collaborating well and intervene to encourage collaboration.

3.1 Behavior Coding Process

Videos of each group’s interactions (Figure 3) were captured by high-angled cameras and synchronized with audio data captured by microphones positioned near each group; additionally, an overhead fisheye lens camera captured the entire class, including events such as the TAs’ interactions with groups. The collected video data were annotated (coded) at the group level by two trained annotators with an annotation scheme adapted from previous work on collaborative behavior annotation [38] to define group activity in terms of task relatedness, peer verbal interaction, TA interaction, talk content, and tablet usage.

Previous research on predicting collaboration from interactions with software has involved annotating similar content

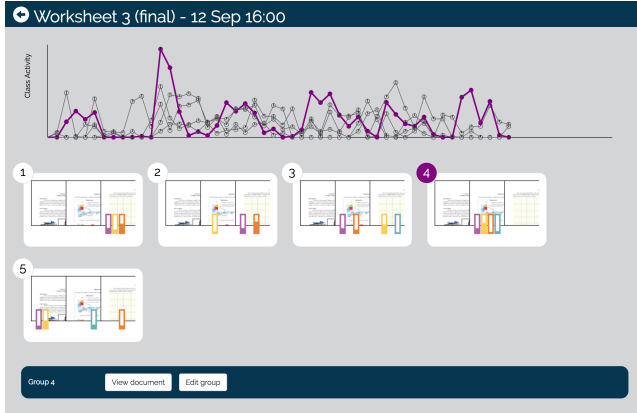


Figure 2: Example screenshot of a teaching assistant’s view of a classroom with five groups of students. The top graph indicates activity over time for each group, with the selected group (#4) highlighted in purple. Bars on each worksheet thumbnail show the page each student is viewing and their individual levels of activity.



Figure 3: An example of a group working collaboratively on a problem through the tool on tablet computers. Videos of such groups were recorded and qualitatively coded through a coding scheme adapted from Paquette et al.’s work [38].

in video clips at 60-second intervals [38]. In this study, the presence of collaborative behaviors (expanded below) were annotated at 20-second video clips, after trials of the annotation process at different clip duration of 10, 20, 30, 40, and 60 seconds. Annotators determined 20 seconds to be a reasonable balance—10 seconds was too brief to confidently observe the presence of collaborative behaviors, while 30 seconds was too long and often led to the observance of multiple collaborative behaviors within the same video clip. Furthermore, through our trials at varied clip lengths, additional identifiable behaviors emerged that were better identified at the current 20-second coding clip length rather than the longer 60-second clips annotated in previous work. A total of 16,270 clips were annotated for the presence (annotated as 1), or absence (0) of the following set of collaborative behaviors:

- *Task relatedness*: At least one of the group members appears to be on task (e.g. two students solving problems on the tablet).
- *Peer verbal interaction*: Verbal interaction is present between group members.
- *TA class interaction*: TA is talking to the whole class (e.g., class-related announcement, addressing a frequently asked question).
- *TA group interaction*: TA is verbally interacting with at least one of the group members.
- *Task talk*: Audible talk content in the group is related to solving the task.
- *Other talk*: Audible talk content in the group is not related to solving the task.
- *Tablet movement*: At least one of the group members is moving the tablet to initiate (and to end) sharing of the screen content with others.

We measured inter-rater reliability via Cohen’s kappa [18] and percent agreement on a subset of 2,125 video clips. Table 1 shows these reliabilities. All labels except *Other talk* (kappa = .651) achieved kappa = .8 or higher, indicating substantial agreement [18]. Given this agreement, the two annotators divided the remaining 14,145 clips and annotated them individually.

Table 1: Inter-rater reliability for a sample of 2,125 video clips in this study.

Behavior	Base rate	Agreement	Kappa
<i>Task relatedness</i>	.954	98.6%	.840
<i>Peer verbal interaction</i>	.501	91.7%	.833
<i>TA class interaction</i>	.024	99.5%	.898
<i>TA group interaction</i>	.150	98.3%	.932
<i>Task talk</i>	.608	91.2%	.816
<i>Other talk</i>	.072	95.3%	.651
<i>Tablet movement</i>	.019	99.2%	.801

Of the qualitatively coded behaviors, we considered six specific behaviors for this study: ON-TASK (derived directly from *Task relatedness*), ON-TASK-NO-INTERACTION (from a combination of *Task relatedness* and *Peer verbal interaction*), PEER-INTERACTION (from *Peer verbal interaction*), SILENT (from *Task talk* and *Other talk*), TASK-TALK (from *Task talk*), and TA-CLASS (from *TA class interaction*). These six behaviors were those which we believed would be best suited for investigating the evolution of collaboration with consideration of the actions of both the TA and students during a typical class period for the course. We did not include tablet movement due to the low base rate during annotation and questionable value for characterizing collaboration. We deemed ON-TASK-NO-INTERACTION important for distinguishing collaboration from individual work, and while it was not explicitly annotated, it was calculated from a combination of two different behavior labels (*Task relatedness* and *Peer verbal interaction*).

3.2 Data Processing

The tablet tool collected student action log files, one per group, during each class session. Relevant behavior data were cleaned and stored based on expected suitability for predicting collaborative behaviors on the tool. These types of data included event types, such as scrolling, drawing, object creation (inserting one of a few built-in graphics), modifying drawings or objects (removing or undoing), as well as the size and position of edits made, object geometry changes (e.g., moving, resizing), page number, scroll bar position, and changes to drawing color.

3.3 Machine Learning Models from Feature Subsets

We aligned annotated behaviors with the student action log files to allow synchronized analysis between the two data sources. We created three features sets: (1) *onset* features, which characterized collaborative behaviors found in and calculated within the first ten minutes, (2) *concurrent* features, which captured collaborative behaviors based on the most recent 60 seconds as well as all cumulative data, and (3) *combined* features, which combines both subsets. Student behaviors were recorded individually within each group's log file. However, we primarily extracted features intended to characterize whole-group behaviors, in line with the group-level video annotation scheme and the overall project goal of improving collaboration rather than individual learning behaviors.

3.3.1 Feature Engineering

Designing features to extract took place over the course of several sessions involving the video annotators and researchers, who discussed behaviors observed in the classroom and how they might be reflected in tablet-based behaviors.

We extracted 89 features from the action logs using the full 50-minutes of the class duration, which we refer to as concurrent features. For these features, we used a combination of the behaviors that annotators had observed to be related to collaboration, as well as those characteristics we hypothesized to be more broadly associated with effective collaboration. For example, we created features such as: the mean distance between consecutive edits of the same students (since it may

distinguish working in one area vs. jumping around rapidly), total number of unique document pages viewed (a higher number may symbolize more exploration of the task), and maximum distance between concurrent edits of the same page but made by different students (may symbolize task division).

Similarly, we extracted 21 features from the action logs calculated from the first ten minutes of class, which we refer to as onset features. Assuming the five stages of small group development apply in this context, we approximately split each 50-minute class period into stages by dividing into fifths. We expected each class period to somewhat reset the collaboration process, since there was a new task each week—meaning a new corresponding task identification stage (storming), as well as some variation in the group, in number and person, due to fluctuating attendance. We specifically kept in mind the characteristics of the task identification stage, such as verbal and written communication for contextualizing the problem and setting “ground rules”, as well as behaviors such as reading or using visual figures to understand (but not necessarily solve) the exercises. To that end, we created features such as: the proportion of the first ten tool objects created by the group being the pre-made available diagrams (a higher proportion may mean more complete solutions early in class), the longest time between object additions and edits (longer pauses between actions may characterize more verbal communication), and the cumulative number of page switches (switching back and forth between pages may signal wanting to fully understand the task at hand by referencing material on other pages).

3.3.2 Machine Learning and Cross-Validation

We used the random forest classifier in the scikit-learn Python library to build models from each respective feature subset [39]. We selected random forest due to its effectiveness in dealing with high dimensional feature spaces, and reducing overfitting [27, 9]. It is also able to deal with highly correlated features, and provides feature importance measurements which we analyzed to find the features that were most predictive of collaboration. We cross-validated models via leave-one-group-out (each of the 20 groups used as the testing set once), and tuned hyperparameters using nested cross-validation and grid search within training data only. Hyperparameters consisted of the proportion of features to consider for each tree branch (0.25, 0.5, 0.75, or 1.0) and the minimum number of instances required in a tree node to create new branches (2, 4, 8, or 16).

Table 4 presents the values of r_{pb} , kappa, and area under the receiver operating characteristic curve (AUC) of the models, cross-validated over all data ignoring the five collaboration phases. We decided to use the point biserial correlation coefficient, r_{pb} , of the true and predicted values as the primary accuracy metric, since the extreme base rates of ON-TASK and TA-CLASS behaviors (and the changing base rates of other behaviors over collaboration phases) led to unwanted sensitivity to the threshold for kappa calculation. Kappa scores (without threshold tuning) were not necessarily representative of accuracy changes as much as poorly-chosen decision thresholds. Table 4 shows that the pattern of AUC values across behavior labels was similar to r_{pb} ; however, r_{pb} allows straightforward computation of confidence intervals, enabling

Table 2: Top ten most important features for each of the six considered collaborative behaviors from the combined features (onset + concurrent) random forest model. Common features in all six behaviors are in bold.

ON-TASK	ON-TASK-NO-INTERACTION	PEER INTERACTION
1. maximum seconds of no actions	1. number of actions	1. cumul. number of page changes
2. cumulative ratio of 2nd most to most active	2. cumul. number of actions	2. cumul. distance drawn
3. cumul. number of page changes	3. cumul. distance drawn	3. cumul. number of actions
4. number of actions	4. maximum seconds of no actions	4. cumul. ratio of 2nd most to most active
5. ratio of least to most active student	5. cumul. mean distance of same student edits	5. cumul. number of scroll position changes
6. proportion of students acting	6. cumul. number of scroll position changes	6. cumul. mean distance of same student edits
7. cumul. number of scroll position changes	7. cumul. ratio of least to most active student	7. cumul. number of tool changes
8. number of unique pages viewed	8. cumul. standard deviation of distance scrolled	8. cumul. ratio of least to most active student
9. max proportion of students on different pages	9. cumul. number of page changes	9. cumul. number of add object
10. cumul. number of actions	10. cumul. ratio of 2nd most to most active	10. cumul. mean y-axis value of edits
SILENT	TASK-TALK	TA-CLASS
1. cumul. number of actions	1. cumul. number of page changes	1. cumul. number of actions
2. cumul. number of selection changes	2. cumul. number of selection changes	2. cumul. standard deviation of distance scrolled
3. cumul. number of add object	3. cumul. distance drawn	3. cumul. number of scroll position changes
4. cumul. distance drawn	4. cumul. number of actions	4. cumul. maximum seconds of no actions
5. cumul. number of page changes	5. cumul. number of add object	5. cumul. proportion of students scrolling
6. cumul. number of tool changes	6. cumul. number of tool changes	6. cumul. number of page changes
7. cumul. mean distance of consecutive edits	7. cumul. number of scroll position changes	7. cumul. distance drawn
8. cumul. ratio of 2nd most to most active	8. cumul. mean distance of same student edits	8. cumul. number of add object
9. maximum seconds of no actions	9. maximum seconds of no actions	9. cumul. distance scrolled
10. cumul. number of scroll position changes	10. cumul. ratio of 2nd most to most active	10. cumul. number of selection changes

the statistical comparisons of models that we include in this paper. We thus proceeded with r_{pb} as the primary accuracy metric.

4. RESULTS

Within a 50-minute class period, we surmised that the five stages of a collaborative problem-solving team could be approximated through five equal 10-minute segments. However, if base rates of each behavior vary over time, model accuracy could as well [29]. Thus, before answering our research questions, we visualized the base rates of each behavior to help inform the results.

4.1 Base Rates Over Time

The trajectories of average base rates of the collaborative learning behaviors across these five segments of class are shown in Figure 4. Across behaviors, we observed a common pattern: the largest changes in base rates were from the first 10-minute segment to the second. The magnitude and direction of the changes in base rates during this transition were variable between the different behaviors, though some patterns can be assumed to be closely correlated. For example, the behaviors TA-CLASS and SILENT followed a similar negative trend in magnitude, since students across groups are more likely to be silent when the TA is addressing the entire class at the start of the class period, when task objectives or announcements are likely to be made. Similarly, ON-TASK, PEER-INTERACTION, and TASK-TALK tended to increase during class periods, since ON-TASK behavior is likely to involve more instances of PEER-INTERACTION and TASK-TALK behavior. ON-TASK-NO-INTERACTION showed a comparatively consistent base rate throughout the class period, perhaps being influenced by other behaviors in both directions with similar magnitude.

As base rates become more imbalanced (closer to 0 or 1),

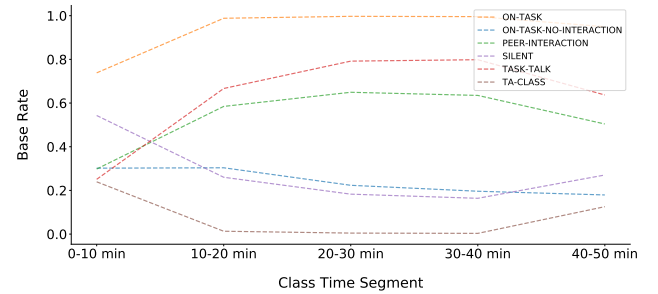


Figure 4: Average base rates of annotated behaviors across segments of each class period (averaged across class periods).

classification problems tend to become more difficult because fewer data points are available from one category of the data, and because accuracy metrics tend to become less effective [29]. Hence, the patterns in Figure 4 are important to consider when interpreting the results of the research questions.

4.2 RQ1: How does the accuracy of predicting collaborative behaviors vary across periods of a class?

To address this research question, we focused on the accuracy of the concurrent features model. This model has similar accuracy to the combined features model (see RQ3), and is more parsimonious since it has 89 features, compared to 110 features from the combined model. Thus, it will likely be the model of choice to drive predictions in future versions of the TA-facing tablet tool, and we focus RQ1 on this model.

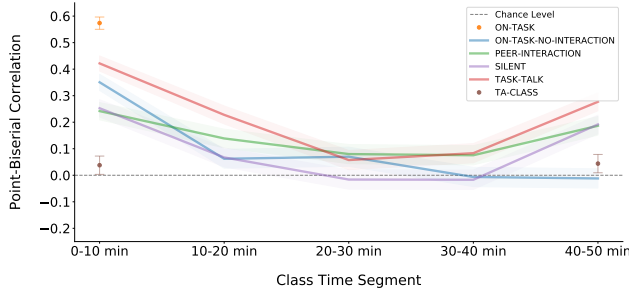


Figure 5: Concurrent features model accuracy shown over time throughout five segments of classes. Accuracy in this case consists of point-biserial correlation coefficients with 95% confidence intervals indicated by shading or by error bars for labels where base rates were too imbalanced (0% or 100%) to allow prediction in every class segment.

From the overview of the model performance in Figure 5, a general U-shape pattern can be observed across the class period, where the second peak in accuracy toward the end of class never quite reached the initial accuracy from the first ten-minute segment. This trend differed for ON-TASK-NO-INTERACTION, which showed a rapid drop after the first 10-minute segment, from 0.351 to 0.062, and did not later increase. Predictions for ON-TASK-NO-INTERACTION and SILENT also briefly dropped below chance level during the second half of the class period.

4.3 RQ2: Can early class collaborative behaviors alone be used to effectively model and predict collaborative behaviors of the entire class period?

As shown in Figure 6, the onset features model had overall lower accuracy across the class periods compared to the concurrent features model (Figure 5). The absence of the U-shaped pattern from the concurrent features model (Figure 5) suggests that the first 10 minutes of collaborative behaviors may have been sufficiently similar to be captured by a model with features created from behaviors from the entire class duration, but that those behaviors were not the same as the last 10 minutes. With the exception of SILENT, predicted behaviors showed a trend toward the lowest accuracy at the end of class. However, PEER-INT remained significantly above chance for the first 30 minutes of class, indicating that groups’ verbal interactions were—to a certain extent—characterized throughout most of the class period by their first 10 minutes of logged behaviors. When compared to the accuracy pattern for the concurrent model, (Figure 5), accuracy dropped below chance level more often, with ON-TASK-NO-INTERACTION, TASK-TALK, and SILENT behaviors predicted at below chance level for the latter half of the class period.

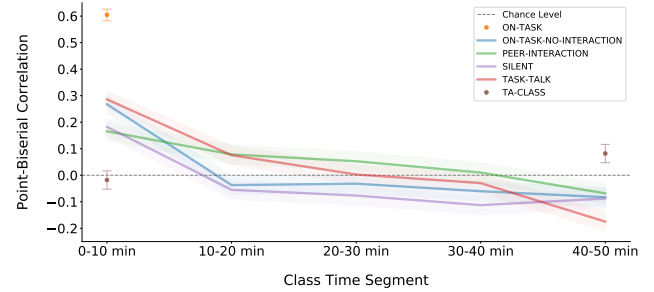


Figure 6: Point-biserial correlation coefficients with 95% confidence intervals by class segment for the onset features model.

4.4 RQ3: Are collaborative behavior prediction models improved through the addition of early class collaborative behavior features, leading to a greater emphasis on the early class period?

The lower and upper 95% confidence interval bounds of the point-biserial correlation values of the models are presented in Table 3. Among the confidence intervals there was overlap for concurrent vs. combined models, but not for onset vs. concurrent and onset vs. combined (with the exception of TA-CLASS for onset vs. concurrent, not drastically so), highlighting that there was no clear significant difference in the models from the addition of onset features to the concurrent features model. This is further supported in Figure 7, which shows that the trajectory of the combined model accuracy closely resembles the concurrent features model (Figure 5). Table 4 also shows that in most cases the combined feature set was not notably better than concurrent features alone when considering overall accuracy across class time segments, in terms of $r_p b$, kappa, or AUC. Feature importance were analyzed and are presented in Table 2. Three common features were found in all six behaviors: *cumulative number of set page*, *cumulative number of scroll position changes*, and *cumulative number of rows*.

5. DISCUSSION

We analyzed automatic detection of collaborative problem solving in classrooms through a lens informed by small group

Table 3: Comparison of the 95% confidence intervals of the models’ point-biserial correlation coefficients, r_{pb}

	Onset	Concurrent	Combined
ON-TASK	.446, .492	.506, .553	.522, .569
ON-TASK-NO-INT	.009, .040	.125, .155	.117, .146
PEER-INT	.075, .104	.247, .276	.220, .250
SILENT	.080, .112	.264, .295	.269, .300
TASK-TALK	.091, .119	.410, .438	.393, .421
TA-CLASS	-.005, .022	.012, .036	.022, .050

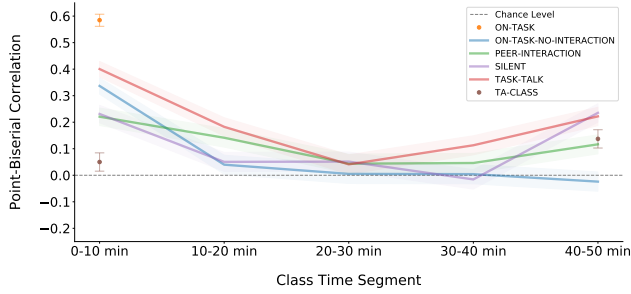


Figure 7: Point-biserial correlation coefficients with 95% confidence intervals by class segment for the combined features model.

development theory. Based on the five stages of small group development and the importance of the early periods of collaboration, we were curious if explicitly considering early class behaviors was beneficial for predicting a team’s collaborative behaviors over a class period. We annotated collaborative behaviors from 16,270 video clips of an undergraduate engineering course as ground truth for behaviors, and compared the accuracy of a machine learning model built from onset features (early collaborative behaviors calculated from the first ten minutes of class) to a model from concurrent features (general collaborative behaviors over the whole class period). In this section, we discuss the implications of our findings.

5.1 Collaboration Across Class Periods

We investigated whether evidence of the five stages of group development could be seen in the concurrent model accuracy when examined in 10-minute periods of a 50-minute class. Our experimental results showed a U-shaped accuracy curve for a majority of the considered collaborative behaviors, with lowest accuracy in the middle 30 minutes of class. The base rate trends (Figure 4) may be one explanation for the observed pattern, because a majority of the behaviors also had U-shaped or inverse U-shaped base rate patterns, which is indicative of unbalanced classes. An exception to the U-shape for accuracy and base rates was ON-TASK-NO-INTERACTION, which had the highest accuracy at the beginning of class and lowest by the end. One possible explanation for this may be that the students in the first 10 minutes of the class were reading or individually thinking about the task, and transitioning to become more verbal and interactive as the class goes on—which can be approximately observed in the overall base rate pattern.

In terms of the small group development theories, the U-shape may be interpreted as evidence for the existence of three, as opposed to five, distinct stages: a beginning, a longer middle, and an end. Three stages is in line with Spitz and Sadock’s three-stage model from observing the training of nursing students [47]. According to the model, stage one is characterized by anxiety-related emotions, such as curiosity and confusion, stage two is a period of trust and cohesiveness, and stage three is disengagement and anxiety about the group conclusion.

It is difficult to determine whether these stages were captured in our analysis, however, since there were some notable dif-

Table 4: Accuracy comparison of the onset features, concurrent features, and combined features (onset + concurrent) models.

Behavior	Model	r_{pb}	Kappa	AUC
ON-TASK	Onset	.470	.469	.737
	Concurrent	.532	.529	.746
	Combined	.547	.545	.754
ON-TASK-NO-INT	Onset	.025	.025	.512
	Concurrent	.192	.140	.551
	Combined	.175	.131	.549
PEER-INT	Onset	.093	.090	.545
	Concurrent	.266	.261	.630
	Combined	.239	.235	.617
SILENT	Onset	.096	.096	.549
	Concurrent	.306	.279	.620
	Combined	.307	.284	.623
TASK-TALK	Onset	.115	.105	.558
	Concurrent	.445	.424	.699
	Combined	.422	.407	.692
TA-CLASS	Onset	.011	.008	.503
	Concurrent	.064	.024	.507
	Combined	.095	.036	.510

ferences in context and aim between our study and Spitz and Sadock’s research. In our study, we did not set out to capture or identify emotions during collaboration, since the focus in data collection was on annotating collaborative behaviors and capturing action data, such as tool use, scrolling, and editing. Furthermore, while previous work has developed approaches for detecting student affect through applying computer vision techniques to detect facial expressions and bodily movements on video [15, 53, 6, 7], our study used video data as means to obtain ground truth data for collaboration rather than emotion. A central goal of our research is to enable analysis for real-time collaboration intervention in the classroom, and thus we analyzed ways to detect collaboration using solely action log data, which can be applied in large and varied classroom environments even when sensors are not available. Current methods for accurately capturing emotion during learning largely rely on video or multimodal methods [5, 20], and it is difficult to envision classrooms with access to multimodal instruments and camera systems designed for analyzing emotion and collaboration.

5.2 Role of Early Collaborative Behaviors

Our hypothesis that early class behaviors could effectively predict the quality of collaboration later in class was not supported by our findings. While we created onset features with characteristics of task identification of problem solving, such as verbal communication, deliberation, and reading, through features such as handwriting on the tablet, pauses between edits, high number of object removals, frequent page switches, and problem diagramming, the accuracy of the onset features model was lower than the concurrent model as a whole. Moreover, the U-shaped pattern from concurrent

features model was not observed. Despite the accuracy of the onset model showing a similarly steep decrease after the first ten minutes, it did not increase at the end of the class for any of the considered collaborative behaviors, as had the concurrent model. Taken together, the U-shape of concurrent model accuracy and the steep decline in accuracy of the onset model suggest that the first ten and last ten minutes of class are similar, but there are differences which make it difficult to effectively characterize based on features calculated from the first ten minutes of class. The similarities of the first and last ten minutes are also supported from the trends in the base rates (Figure 4). TA-CLASS behaviors—when the instructor addresses the entire class—only tend to occur at the beginning and end of class, but the content of the announcements at the beginning of the class are different and likely influence student behavior differently. For example, students may be more likely to listen and be silent in response to the announcements made at the beginning of the class since it is immediately pertinent to the class ahead, but students may be less silent when announcements are made at the class end.

Our analysis also did not support the idea that the addition of the onset features (21 features) to the concurrent features (89 features) model might improve predictive accuracy. While the resulting combined model was created using the largest number (110) of features with an emphasis on the earlier parts of class, the accuracy did not differ significantly from the concurrent features model. Comparing the confidence intervals of the model’s overall point-biserial correlation coefficient (Table 3) showed that while the accuracy of onset and concurrent features models are significantly different for a majority of the behaviors except TA-CLASS (which has especially imbalanced base rates), there is overlap between concurrent and combined models for all behaviors. This indicates that the models may not be statistically different, and are not meaningfully different. Moreover, of the 110 features in the combined (onset + concurrent) model, none of the 21 onset features were found in the top ten important features in any of the six behaviors (Table 2). Three common features were found in the top ten important features for all six behaviors: *cumulative number of set page*, *cumulative number of scroll position changes*, and *cumulative number of rows*. When interpreted together, these three features may be related to the overall activity level of the groups, which may intuitively relate to changes in collaborative behavior.

6. CONCLUSION

In this paper, we were motivated by theories in small group development to analyze how explicitly accounting for early class behaviors and collaboration evolution might help improve collaboration prediction from tool action log data. We investigated collaborative problem solving in an introductory engineering course over 12 weeks. We found that collaboration prediction in a 50-minute class period did not appear to follow a straightforward interpretation of the five-stage structure, but rather a potential three-stage structure. We found that while the first ten minutes of class are distinct from the middle and ending periods of class, onset features calculated from the first ten minutes of the class could not be used to effectively predict collaboration in the later parts of class. Concurrent features (calculated from the whole 50-minute period) performed better as a whole, and the combination

of onset and concurrent features did not necessarily lead to a better predicting model. Thus, groups’ collaborative behaviors later in class were not notably related to their initial collaborative behaviors.

Our study was limited in several ways. Using solely tablet action log data to examine small group development restricted us from being able to account for changes in emotion, a common aspect of small group development theory. We utilized data from action logs since we wanted our analysis to be scalable and make progress toward real-time student interventions for collaboration in the classroom via prompts delivered to TAs (Figure 2). However, to promote better understanding collaborative learning theory in general, additional approaches are needed. To this end, future work examining small group development in collaborative problem solving may benefit from incorporating work on sensor-free affect detection for student engagement [2, 30], which may help identify emotions associated with various stages of small group development such as confusion or anxiety [47]. Additionally, audio of group conversations could be recorded from their tablets and aligned with action log data to understand conversations in the context of the small group development at hand. Our study was also limited by the variability of student groups in size and membership. Some groups had as few as two students in some weeks of class, and the same groups may have had four members in other weeks. This likely influenced the amount of activity captured, in addition to inevitable changes in the communication dynamic.

Insights into the influence of early collaborative behaviors for improving collaboration prediction may help design better interventions for helping TAs facilitate collaboration, and design software tool features to promote effective student collaboration. Deeper insights into understanding small group evolution may offer ways for future work to more accurately identify a group’s current collaborative stage solely from a group’s behaviors, without considering content of interactions between members. Based on the assumed stage, instructors or tools could possibly allow for personalized per-team interventions to better facilitate collaborative problem solving.

7. ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Grant No. 1441149 and 1628976. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

- [1] J. Andrews-Todd, C. Forsyth, J. Steinberg, and A. Rupp. Identifying profiles of collaborative problem solvers in an online electronics environment. *International Educational Data Mining Society*, 2018.
- [2] R. S. Baker, S. M. Gowda, M. Wixon, J. Kalka, A. Z. Wagner, A. Salvi, V. Aleven, G. W. Kusbit, J. Ocumpaugh, and L. Rossi. Towards sensor-free affect detection in cognitive tutor algebra. *International Educational Data Mining Society*, 2012.
- [3] J. Barr and A. Gunawardena. Classroom salon: a tool for social collaboration. In *Proceedings of the 43rd*

- [4] B. Barron. Achieving coordination in collaborative problem-solving groups. *The journal of the learning sciences*, 9(4):403–436, 2000.
- [5] N. Bosch and S. K. D’Mello. The affective experience of novice computer programmers. *International journal of artificial intelligence in education*, 27(1):181–206, 2017.
- [6] N. Bosch, S. K. D’Mello, R. S. Baker, J. Ocumpaugh, V. Shute, M. Ventura, L. Wang, and W. Zhao. Detecting student emotions in computer-enabled classrooms. In *IJCAI*, pages 4125–4129, 2016.
- [7] N. Bosch, S. K. D’Mello, J. Ocumpaugh, R. S. Baker, and V. Shute. Using video to automatically detect learner affect in computer-enabled classrooms. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 6(2):1–26, 2016.
- [8] L. J. Braaten and B. LJ. Development phases of encounter groups and related intensive groups. a critical review of models and a new proposal. *Interpersonal Development*, 1974.
- [9] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [10] D. Brijlall. Exploring the stages of polya’s problem-solving model during collaborative learning: A case of fractions. *International Journal of Educational Sciences*, 11(3):291–299, 2015.
- [11] K. A. Bruffee. *Collaborative learning: Higher education, interdependence, and the authority of knowledge*. ERIC, 1999.
- [12] P. S. Buffum, M. Frankosky, K. E. Boyer, E. N. Wiebe, B. W. Mott, and J. C. Lester. Mining sequences of gameplay for embedded assessment in collaborative learning. In *EDM*, pages 575–576. ERIC, 2016.
- [13] J.-M. Burkhardt, F. D tienne, A.-M. H bert, L. Perron, S. Safin, and P. Leclercq. An approach to assess the quality of collaboration in technology-mediated design situations. In *Proceedings of ECCE 2009: European Conference on Cognitive Ergonomics*, 2009.
- [14] Z. Cai, B. Eagan, N. Dowell, J. Pennebaker, D. Shaffer, and A. Graesser. Epistemic network analysis and topic modeling for chat data from collaborative learning environment. In *Proceedings of the 10th international conference on educational data mining*, 2017.
- [15] R. A. Calvo and S. K. D’Mello. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on affective computing*, 1(1):18–37, 2010.
- [16] R. B. Caple. The sequential stages of group development. *Small Group Behavior*, 9(4):470–76, 1978.
- [17] K. Cassidy. Tuckman revisited: Proposing a new model of group development for practitioners, 2007.
- [18] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.
- [19] P. Dillenbourg. What do you mean by collaborative learning?, 1999.
- [20] S. K. D’Mello, N. Bosch, and H. Chen. Multimodal-multisensor affect detection. In *The Handbook of Multimodal-Multisensor Interfaces: Signal Processing, Architectures, and Detection of Emotion and Cognition-Volume 2*, pages 167–202. ACM, 2018.
- [21] D. C. Dunphy. Phases, roles, and myths in self-analytic groups. *The Journal of Applied Behavioral Science*, 4(2):195–225, 1968.
- [22] J. Garland, H. Jones, and R. Kolodny. A model for stages of development in social work groups. *Explorations in group work*, pages 17–71, 1965.
- [23] A. A. Gokhale. Collaborative learning enhances critical thinking. *Journal of Technology Education*, 1995.
- [24] R. Goldman, R. Pea, B. Barron, and S. J. Derry. *Video research in the learning sciences*. Routledge, 2014.
- [25] P. Griffin and E. Care. *Assessment and teaching of 21st century skills: Methods and approach*. Springer, 2014.
- [26] F. Hesse, E. Care, J. Buder, K. Sassenberg, and P. Griffin. A framework for teachable collaborative problem solving skills. In *Assessment and teaching of 21st century skills*, pages 37–56. Springer, 2015.
- [27] T. K. Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.
- [28] K. Huang, T. Bryant, and B. Schneider. Identifying collaborative learning states using unsupervised machine learning on eye-tracking, physiological and motion sensor data. In *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, volume 318, page 323. ERIC, 2019.
- [29] L. A. Jeni, J. F. Cohn, and F. De la Torre. Facing imbalanced data—Recommendations for the use of performance metrics. In *Proceedings of the 5th International Conference on Affective Computing and Intelligent Interaction*, pages 245–251, Sept. 2013.
- [30] Y. Jiang, N. Bosch, R. S. Baker, L. Paquette, J. Ocumpaugh, J. M. A. L. Andres, A. L. Moore, and G. Biswas. Expert feature-engineering vs. deep neural networks: which is better for sensor-free affect detection? In *International Conference on Artificial Intelligence in Education*, pages 198–211. Springer, 2018.
- [31] J. Kang, D. An, L. Yan, and M. Liu. Collaborative problem-solving process in a science serious game: Exploring group action similarity trajectory. In *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*. ERIC, 2019.
- [32] M. Kapur. Temporality matters: Advancing a method for analyzing problem-solving processes in a computer-supported collaborative environment. *International Journal of Computer-Supported Collaborative Learning*, 6(1):39–56, 2011.
- [33] M. A. Kazemitabar, S. Bodnar, P. Hogaboam, Y. Chen, J. P. Sarmiento, S. P. Lajoie, C. Hmelo-Silver, R. Goldman, J. Wiseman, and L. Chan. Creating instructor dashboards to foster collaborative learning in on-line medical problem-based learning situations. In *International Conference on Learning and Collaboration Technologies*, pages 36–47. Springer, 2016.
- [34] A. Lieberman. Collaborative research: Working with, not working on. *Educational leadership*, 43(5):28–32, 1986.
- [35] E. Mercier, S. Shehab, J. Sun, and N. Capell. The development of collaborative practices in introductory engineering courses. In *Exploring the Material Conditions of Learning: Computer Supported*

- Collaborative Learning (CSCL) Conference*, pages 657–658, 2015.
- [36] L. R. Offermann and R. K. Spiros. The science and practice of team development: Improving the link. *Academy of Management Journal*, 44(2):376–392, 2001.
 - [37] J. K. Olsen, V. Aleven, and N. Rummel. Predicting student performance in a collaborative learning environment. *International Educational Data Mining Society*, 2015.
 - [38] L. Paquette, N. Bosch, E. Mercier, J. Jung, S. Shehab, and Y. Tong. Matching data-driven models of group interactions to video analysis of collaborative problem solving on tablet computers. In *Proceedings of International Conference of the Learning Sciences, ICLS*, pages 312–319. International Conference of the Learning Sciences, June 2018.
 - [39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, Nov. 2011.
 - [40] J. M. Reilly, M. Ravenell, and B. Schneider. Exploring collaboration using motion sensors and multi-modal learning analytics. *International Educational Data Mining Society*, 2018.
 - [41] J. M. Reilly and B. Schneider. Predicting the quality of collaborative problem solving through linguistic analysis of discourse. In *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, pages 149–157. ERIC, 2019.
 - [42] P. Reimann. Time is precious: Variable-and event-centred approaches to process analysis in cscl research. *International Journal of Computer-Supported Collaborative Learning*, 4(3):239–257, 2009.
 - [43] F. J. Reynolds and R. A. Reeve. Gesture in collaborative mathematics problem-solving. *The Journal of Mathematical Behavior*, 20(4):447–460, 2001.
 - [44] J. Roschelle and S. D. Teasley. The construction of shared knowledge in collaborative problem solving. In *Computer supported collaborative learning*, pages 69–97. Springer, 1995.
 - [45] P. H. Sins, W. R. Van Joolingen, E. R. Savelsbergh, and B. van Hout-Wolters. Motivation and performance within a collaborative computer-based modeling task: Relations between students’ achievement goal orientation, self-efficacy, cognitive processing, and achievement. *Contemporary Educational Psychology*, 33(1):58–77, 2008.
 - [46] W. M. Smith. Observations over the lifetime of a small isolated group: Structure, danger, boredom, and vision. *Psychological reports*, 19(2):475–514, 1966.
 - [47] H. Spitz and B. Sadock. Psychiatric training of graduate nursing students. use of small interactional groups. *New York state journal of medicine*, 73(11):1334–1338, 1973.
 - [48] J. Thannhauser, S. Russell-Mayhew, and C. Scott. Measures of interprofessional education and collaboration. *Journal of interprofessional care*, 24(4):336–349, 2010.
 - [49] B. W. Tuckman. Developmental sequence in small groups. *Psychological bulletin*, 63(6):384, 1965.
 - [50] B. W. Tuckman and M. A. C. Jensen. Stages of small-group development revisited. *Group & Organization Studies*, 2(4):419–427, 1977.
 - [51] M. D. Weber and T. A. Karman. Student group approach to teaching using tuckman model of group development. *Advances in Physiology Education*, 261(6):S12, 1991.
 - [52] M. Yee-King and M. d’Inverno. Stimulating collaborative activity in online social learning environments with markov decision processes. In *EDM*, pages 652–653, 2016.
 - [53] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE transactions on pattern analysis and machine intelligence*, 31(1):39–58, 2008.
 - [54] L. A. Zurcher Jr. Stages of development in poverty program neighborhood action committees. *The Journal of Applied Behavioral Science*, 5(2):223–258, 1969.

Evaluating sources of course information and models of representation on a variety of institutional prediction tasks

WeiJie Jiang
University of California, Berkeley
jiangwj@berkeley.edu

Zachary A. Pardos
University of California, Berkeley
zp@berkeley.edu

ABSTRACT

Data mining of course enrollment and course description records has soared as institutions of higher education begin tapping into the value of these data for academic and internal research purposes. This has led to a more than doubling of papers on course prediction tasks every year. The papers often center around a single prediction task and introduce a single novel modeling approach utilizing one or two data sources. In this paper, we provide the most comprehensive evaluation to date of data sources, models, and their performance on downstream prediction tasks. We separately incorporate syllabus, catalog description, and enrollment history data to represent courses using graph embedding, course2vec (i.e., skip-gram), and classic bag-of-words models. We evaluate these representations on the tasks of predicting course prerequisites, credit equivalencies, student next semester enrollments, and student course grades. Most notably, our results show that syllabi bag-of-words representations performed better than course descriptions in predicting prerequisite relationships, though enrollment-based graph embeddings performed substantially better still. Course descriptions provided the highest single representation accuracy in predicting course similarity, with descriptions, syllabi, and course2vec combined representations providing the highest ensembled accuracy on this task.

Keywords

Higher education, course recommendation, course2vec, prerequisites, enrollment histories, syllabus, network embedding, grade prediction, institutional analytics.

1. INTRODUCTION

Data from institutions of higher education are quickly coming into focus for educational data mining and learning analytics communities as the utility of these data start to become clear and attention begins to shift from the informal learning context of free online courses to the higher stakes context of degree granting institutions and their students.

WeiJie Jiang and Zach Pardos "Evaluating sources of course information and models of representation on a variety of institutional prediction tasks" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 115 - 125

Educational Data Mining (EDM) plays an important role in the developing stages of methodological adaptation to a domain by evaluating new sources of data for their utility in existing models and tasks and updating the utility of existing data as models and tasks evolve. Recently, EDM has seen a more than doubling year-to-year in papers focused on prediction with large institutional enrollment sets from the formal higher education context, with a single paper on the topic in 2017 [38], two in 2018 [12, 6], and five in 2019 [29, 36, 19, 37, 16], though early pioneering work on predicting academic outcomes date back to the first EDM conference [39, 2].

In this paper, we summarize and evaluate this quickly developing domain across three dimensions: sources of institutional data, models for representing students and courses, and the performance of the former two categories on institutionally relevant prediction tasks. As academic researchers and practitioners know, not all sources of data are always available and different costs are associated with obtaining a new source. Similarly, when it comes to modeling, different personnel and computational costs are associated with applying models depending on their complexity and recency of introduction. We provide the most comprehensive evaluation to date of the performance of different combinations of data and models on common institutional tasks emerging in the literature so that the costs and benefits of each, in our setting, can be quickly appraised. In addition to evaluating previously introduced approaches and data, we introduce large scale syllabus data as a novel source of information about courses and a novel application of a nascent graph-embedding approach for representing courses.

2. RELATED WORK

Contemporary approaches to data mining institutional datasets in higher education have distinguished themselves from earlier drop-out detection work [18] in the use enrollment data and adoption of representational methods that factorize, embed, or otherwise vectorize courses into a space. This began with [10] that used matrix factorization applied to student enrollments and observed that the factorization grouped courses and students in semantically meaningful ways. Subsequent research also employed matrix factorization for grade prediction tasks [38, 37]. Neural embedding models followed, with the skip-gram neural network model applied to sequences of course enrollments, an approach coined "Course2vec" [32]. The course embeddings extracted from this model were found to be predictive of on-

Table 1: Related work on institutional prediction tasks (columns) and sources of data used in the task (rows)

	Grade prediction	Enrollment prediction	Prerequisite prediction	Course similarity
Course grades	[10, 15, 21, 38, 37, 16]	[10, 3, 32, 36]	[22, 11, 21, 16]	[17, 25, 12, 29]
Enrollment histories	[10, 15, 21, 38, 37, 16]	[10, 3, 32, 36, 1]	[22, 11, 21, 16]	[17, 25, 31, 33, 29]
Major declarations	[38, 21, 37]	[32, 36]	[21]	[25]
Catalog descriptions		[32]		[26, 31, 33, 12, 29]

time graduation [25], course similarity within [33] and across institutions [31], and of latent topics of courses [8]. Student course selections have also been posed as a graph, treating courses as nodes and student course selections as strengthening the edges between courses the more frequently they share students in common [15, 16, 1]. The aforementioned approaches all use student course selections, a collaborative signal, to represent a course. Other approaches utilize content data of a course (e.g., catalog description) for representation and for downstream tasks such as course similarity analysis [26, 31, 33, 12, 29] and enrollment prediction [32]. Several papers have collected course ratings for modeling and recommendation [13, 12].

The majority of models in related works have been framed as potentially contributing to a course recommendation system, or already integrated into one. They commonly focused on grade prediction [10, 15, 21, 38, 37, 16] as a necessary first-step towards a preparation, or goal-based [21] recommendation system that could aid students in preparing for difficult courses. In a similar vein, prerequisite course inference has been framed [22, 11, 21, 16] also as a potential means to help guide students towards course taking paths expected to be more successful than others [11, 30]. Table 1 summarizes this body of work in terms of the most common data sources used (i.e., course grades, enrollment histories, major declarations, and catalog descriptions) and most common evaluation tasks (i.e., grade prediction, enrollment prediction, prerequisite prediction, and course similarity) focused on in this paper.

3. DATA SOURCES

In this section, we will describe the three primary sources of data utilized in this paper. First, we will describe the source generally, followed by a paragraph detailing the particulars of the dataset used in our offline evaluation experiments.

3.1 Enrollment histories and grades

A student’s transcript is classically a report containing the student’s histories of courses taken and the grade achieved in each. Enterprise database systems often store raw forms of these data. It has become more common for institutions to not only store these data in relational form but for their internal offices of institutional analytics to have ready access to them. As the fields of EDM and learning analytics have grown, these data have become more available to faculty to aid scholarly research. We used an anonymised enrollments and grades dataset containing student enrollment histories at a large public university, UC Berkeley, collected from Fall 2008 through Fall 2017. The dataset consists of per-semester (i.e., Fall, Spring, and Summer) class enrollments for 164,196 students (both undergraduates and graduates) with a total of 4.8 million class enrollments. A class enrollment record in the data indicates that the student was still enrolled in the class at the end of the semester. The action of drop-

ping a class is not contained in these data. The median number of classes enrolled by a student in a semester was four. There were 9,478 unique lecture courses from 214 departments hosted in 17 different Divisions of 6 different Colleges. Course meta-information was also included in these data and contained course number, department name, class instructor(s), and room max capacity. In this paper, we only consider lecture courses with at least 20 enrollments total over the 9-year period, resulting in 7,487 courses. Although courses can be categorized as undergraduate courses and graduate courses, undergraduates are allowed to enroll in many of the graduate courses. Enrollment data were sourced from the campus’ enterprise data warehouse.

3.2 Course catalog descriptions

A paper catalog use to be the primary way in which students could browse all the course offerings at an institution. Fortunately, this has been superseded by online catalogs, most of which are searchable. The catalog contains course numbers, their hosting department, and typically a paragraph or type description of the course. Our dataset contains the most recent catalog description of every course in our enrollment histories. The average catalog description length was 325 words with 489 courses having exceptionally short descriptions of 10 words or fewer. We sourced these descriptions from the campus Office of the Registrar official API for Course information. These descriptions were pre-processed by (1) removing generic, often-seen sentences across descriptions (2) removing stop words (3) removing punctuation, and (4) word lemmatization and stemming.

3.3 Course syllabi from the Learning Management System

A course syllabus is a detailed, chronological list of subjects and assignments that a course will cover, often with other logistical information about course meeting place and time and grading policies. While the syllabus is perhaps an ideal source of information to utilize for content-based representation of a course, it has been an elusive source to conduct research on. This is because few institutions mandate that instructors make their syllabi public and therefore it is uncommon to have syllabi centrally stored by the institution to subsequently make available to researchers. An additional barrier to research availability is that many institutions view a syllabus as an instructor’s intellectual property (IP), and therefore not sharable in original form without permission. Our study introduces syllabus data into contemporary predictive models and tasks, but with a caveat that maintains instructor control over the original intellectual property.

The university from which our syllabus data come from considers syllabi to be instructor IP and does not collect them centrally. However, a common place in which instructors often place their syllabi is the “Syllabus” page of the cam-

pus Learning Management System (LMS). We worked with the campus technology services organization in charge of the LMS to extract all text from the Syllabus pages of all courses. Sometimes this page would contain only a link to the pdf of a syllabus, in which case that link was downloaded and parsed to text. To abide by the IP restrictions around course syllabi and respect instructor ownership of them, a workaround was arranged. Only the technology services would have access to the cleanly parsed data from the LMS. They would then pre-process the syllabus themselves, similar to how we pre-processed catalog descriptions, parsing out html, converting it into bag-of-words (BOW) form. This form would thereby make the syllabus unusable as an instructional object but potentially usable by an algorithm attempting to extract information for institutional prediction tasks. It was also agreed that the BOW we received would not be made public and these data could be revoked at any time. There were 3,645 unique courses that contained HTML on the LMS Syllabus page, not including a link to a file. There were 2,712 courses that contained a link to a file, with some courses having both. The total number of courses with some amount of syllabus data was 4,017 with a combined vocabulary of 17,194 unique words.

4. REPRESENTATION MODELS

We choose four approaches of increasing complexity for representing courses. These four reflect the most common paradigms of modeling found in our literature review. The simplest is a content-based bag-of-words representation of the course. The BOW approach could be applied to the catalog description or syllabus of a course, where available. Next is the use of a recently published variant on Course2vec called multifactor Course2vec, which applies a skip-gram to sequences of course enrollments. In addition to embedding courses, multifactor Course2vec also embeds the instructor of the course and the course's department, both presented to the model in the form of a one-hot encoding. Multifactor Course2vec has been shown to perform better on course similarity tasks than the original Course2vec [33], in theory because it separates out factors, such as instructor and department, allowing the course embedding to more purely represent the content. Long Short-Term Memory models are the third model used to embed courses, followed by a recently introduced network embedding technique.

A summary of the approaches used is visually illustrated in Figure 2. The various types of information these methods leveraged are summarized in Table 2.

Table 2: Summary of representative learning methods for courses

	catalog descriptions	course syllabus	course meta-information	enrollment histories	course grades	model type
bag-of-words	✓	✓				static
multi-c2v			✓	✓		dynamic
LSTM				✓		dynamic
sc-AMHEN				✓	✓	static

4.1 Bag-of-words

The basic representation mode of bag-of-words was proposed by information retrieval researchers for text corpora. It is a model that reduces each document in a corpus to a vector of real numbers, each of which represents a term, or vocabulary weight. The term weight can be term frequency, a binary value with 1 indicating that the term occurred in the document and 0 indicating that it did not, or a tf-idf scheme[7]. There are two sources of texts that can represent the content of courses: the course catalog descriptions and course syllabi.

4.2 Multifactor Course2vec

The Course2vec model [32] was proposed to learn distributed representations of courses from students' enrollment records throughout semesters by using a notion of an enrollment sequence as a "sentence" and courses within the sequence as "words", borrowing terminology from the natural language domain. For each student, their chronological course enrollment sequence is produced by first sorting by semester then randomly serializing within-semester course order. Each course enrollment sequence is then trained on like a sentence using a skip-gram model.

More features of courses (e.g., course instructor and department) can be added to the input of the multifactor Course2vec model to enhance the classifier and its representations. The model learns both course and added feature representations by maximizing the objective function over all the students' enrollment sequences and the features of courses, defined as follows.

$$\sum_{s \in S} \sum_{c_i \in s} \sum_{-w < j < w, j \neq 0} \log p(c_{i+j} | c_i, f_{i1}, f_{i2}, \dots, f_{ih}) \quad (1)$$

Probability $p(c_{i+j} | c_i, f_{i1}, f_{i2}, \dots, f_{ih})$ of observing a neighboring course c_{i+j} in window size w given the current course c_i and its features $f_{i1}, f_{i2}, \dots, f_{ih}$ (e.g., instructors, department) can also be defined via the softmax function,

$$p(c_{i+j} | c_i, f_{i1}, f_{i2}, \dots, f_{ih}) = \frac{\exp(\mathbf{a}_i^T \mathbf{v}'_{i+j})}{\sum_{k=1}^n \exp(\mathbf{a}_i^T \mathbf{v}'_k)} \quad (2)$$

$$\mathbf{a}_i = \mathbf{v}_i + \sum_{j=1}^h \mathbf{W}_{n_j \times v} \mathbf{f}_{ij} \quad (3)$$

where \mathbf{a}_i is the vector sum of input course vector representation \mathbf{v}_i and all the features vector representations of course c_i , \mathbf{f}_{ij} is the multi-hot input of the j -th feature of course i , and $\mathbf{W}_{n_j \times v}$ is the weight matrix for feature j . So by multiplying $\mathbf{W}_{n_j \times v}$ and \mathbf{f}_{ij} , it gets the sum of feature vector representations of the i -th course. The illustration of the model is shown in the multi-course part of Figure 2. \mathbf{v}_i is the course representation of course i learned from the model that is used in various down-stream course prediction tasks.

4.3 LSTM-learned Representations

In previous work [32], an LSTM was designed to recommend courses for students to take in the next semester, based on their enrollment histories. The input of the model in each time slice is a multi-hot vector representing the courses taken in the corresponding semester. The weights of the input \mathbf{W}_f , \mathbf{W}_i , \mathbf{W}_o , and \mathbf{W}_c learned by the LSTM transferred the

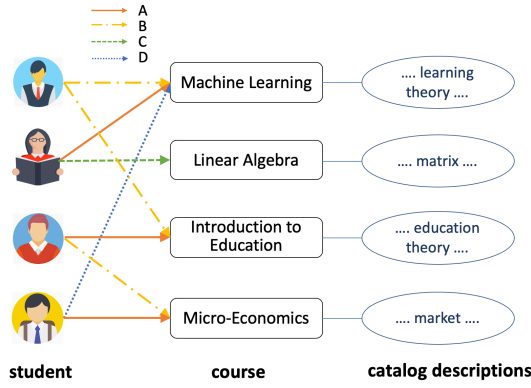


Figure 1: Illustration of the Attributed Multiplex Heterogeneous Network (AMHEN) of Students and Courses.

multi-hot input to the forget gate, input gate, output gate, and the cell in the LSTM cell, respectively. These four sets of weights are combined to form representations of courses that can be used in down-stream prediction tasks.

4.4 Attributed Multiplex Heterogeneous Network Embeddings

Network representation learning (i.e., network embedding), is a promising method to project nodes in a network onto a low-dimensional continuous space while preserving network structure and inherent properties. In terms of the network topology (homogeneous or heterogeneous) and attributed property (with or without attributes), six different types of networks can be categorized, i.e., HOMogeneous Network (HON) [34], Attributed HOMogeneous Network (AHON) [40], HETerogeneous Network (HEN) [9], Attributed HETerogeneous Network (AHEN) [5], Multiplex HETerogeneous Network (MHEN) [24], and Attributed Multiplex HETerogeneous Network (AMHEN) [4]. In the university setting, students and courses can be mapped into a large heterogeneous network, where students and courses are two types of nodes connected by students' enrollments in courses. The proximities between students and courses vary based on the grades (e.g., A, B, C, D, etc.) students received for courses, yielding the network with multiple views, i.e., multiplex heterogeneous network. Furthermore, if we incorporate the attributes of students and nodes (e.g., course catalog descriptions), the network will turn to an Attributed Multiplex HETerogeneous Network (AMHEN), which is illustrated in Figure 1. Because students may receive different grades for the courses they enrolled, we consider different grades as different edge types between students and courses.

DEFINITION 1. (*Attributed Multiplex Heterogeneous Network*): An attributed multiplex heterogeneous network is a network $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, $\mathcal{E} = \cup_{r \in \mathcal{R}} \mathcal{E}_r$, where \mathcal{E}_r consists of all edges with edge type $r \in \mathcal{R}$, and $|\mathcal{R}| > 1$. We separate the network for every edge type $r \in \mathcal{R}$ as $G_r = (\mathcal{V}, \mathcal{E}_r, \mathcal{A})$. Each node $v_i \in \mathcal{V}$ is associated with some types of feature vectors. $A = \{x_i | v_i \in \mathcal{V}\}$ is the set of node features for all nodes, where x_i is the associated node feature of node v_i .

In the student-course attributed multiplex heterogeneous network we described above, $\mathcal{V} = (\mathcal{C}, \mathcal{S})$, where each node $c \in \mathcal{C}$ represents a course in the course set \mathcal{C} and each node $s \in \mathcal{S}$ represents a student in the student set \mathcal{S} . \mathcal{R} refers to all the edge types in the student-course attributed multiplex heterogeneous network, i.e., grade types. As students have enrollment and grade histories of multiple courses, we consider student embeddings as a state of their course knowledge. Different grade types mirror different levels of course knowledge, thus should be represented as different embeddings.

Given the above definitions and descriptions, we can formally define our problem for representation learning on the student-course AMHEN.

PROBLEM 1. (*Student-Course AMHEN Embedding*). Given a Student-Course AMHEN $G = (\mathcal{C}, \mathcal{S}, \mathcal{E}, \mathcal{A})$, the problem of Student-Course AMHEN embedding is to give a unified low-dimensional space representation of each student node $s \in \mathcal{S}$ and each course node $c \in \mathcal{C}$ on every grade type r . The goal is to find a function $g : \mathcal{S} \rightarrow \mathbb{R}^d$ and a function $f_r : \mathcal{C} \rightarrow \mathbb{R}^d$ for every grade (edge) type r , where $d \ll |\mathcal{C}|$ ($d \ll |\mathcal{S}|$).

4.4.1 Student and Course Representations

In this section, we detail our adaptation of the AMHEN framework[4] to the student-course scenario to learn graph-based student and course representations. We split the overall course embedding on each course type r into three parts: base embedding \mathbf{b}_c , grade embedding \mathbf{g} , and attribute embedding \mathbf{u} , and split the overall student embedding into two parts: base embedding \mathbf{b}_s , and individual embedding \mathbf{p} .

The base embedding of course node c_i , i.e., \mathbf{b}_{c_i} , is shared between different grade types. We define \mathbf{b}_{c_i} as a parameterized function of c_i 's attributes $\mathbf{x}_i \in \mathbb{R}^x$ as:

$$\mathbf{b}_{c_i} = h(\mathbf{x}_i) \quad (4)$$

where h is a transformation function, such as a multi-layer perceptron. The attribute embedding of course node c_i , i.e., \mathbf{u}_i , is defined as:

$$\mathbf{u}_i = D^T \mathbf{x}_i \quad (5)$$

Given that in the Student-Course AMHEN, the neighbors of a course are all students while the neighbors of students are all courses, the k -th level¹ of grade embedding $\mathbf{g}_{ir}^{(k)} \in \mathbb{R}^d$, ($1 \leq k \leq K$) of course node c_i on grade type r is aggregated from individual embeddings of students that are c_i 's neighbors, which means these students all received grade type r for course c_i .

$$\mathbf{g}_{ir}^{(k)} = \text{mean}(\{\mathbf{p}_j^{(k-1)}, \forall p_j \in \mathcal{N}_i\}) \quad (6)$$

Similarly, the k -th level of individual embedding $\mathbf{p}_i^{(k)} \in \mathbb{R}^d$, ($1 \leq k \leq K$) of a student node s_i is aggregated from grade embeddings of courses that are s_i 's neighbors, which demonstrates a student's representation is derived from the grade histories of his/her enrolled courses.

$$\mathbf{p}_i^{(k)} = \text{mean}(\{\mathbf{g}_{jr}^{(k-1)}, \forall c_j \in \mathcal{N}_i\}) \quad (7)$$

¹By level we mean iteration, i.e., the embedding is updated after each parameters update process.

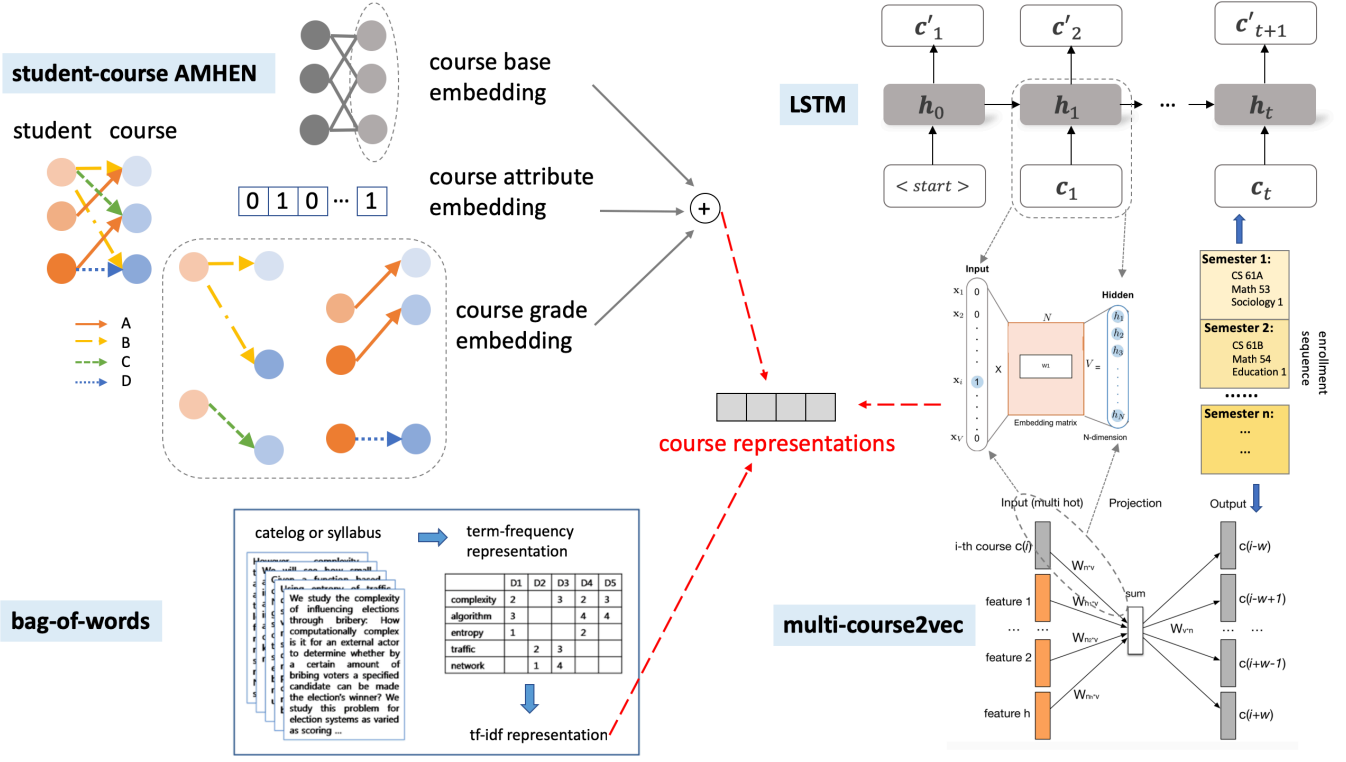


Figure 2: Visual summary of representation learning methods

We denote the k -th level grade embedding $\mathbf{g}_{ir}^{(k)}$ as grade embedding \mathbf{g}_{ir} , and concatenate all the grade embeddings for course node c_i as $\mathbf{G}_i \in \mathbb{R}^{d \times m}$, where d is the dimension of grade embeddings and m is the number of grade types.

$$\mathbf{G}_i = (\mathbf{g}_{i1}, \mathbf{g}_{i2}, \dots, \mathbf{g}_{im}) \quad (8)$$

We use self-attention mechanism[23] to compute the coefficients $\mathbf{a}_{ir} \in \mathbb{R}^m$ of linear combination of vectors in \mathbf{G}_i on edge type r as:

$$\mathbf{a}_{ir} = \text{softmax}(\mathbf{w}_r^T \tanh(\mathbf{W}_r \mathbf{G}_i))^T \quad (9)$$

where $\mathbf{w}_r \in \mathbb{R}^{d_a}$ and $\mathbf{W}_r \in \mathbb{R}^{d_a \times d}$ are trainable parameters for grade type r . Thus, the overall embedding of course node c_i for grade type r is:

$$\mathbf{c}_{ir} = \alpha_c h(\mathbf{x}_i) + \mathbf{M}_r^T \mathbf{G}_i \mathbf{a}_{ir} + \beta_c \mathbf{D}^T \mathbf{x}_i \quad (10)$$

where $\mathbf{M}_r \in \mathbb{R}^{d \times n}$ and $\mathbf{D} \in \mathbb{R}^{x \times n}$ are trainable transformation matrix. α_c and β_c are two coefficients adjusting the weights of the three embeddings of courses, which can also be trainable.

The overall embedding of student node s_i is:

$$\mathbf{s}_i = \alpha_s \mathbf{b}_s + \mathbf{N}^T \mathbf{p}_i \quad (11)$$

where α_s is a trainable coefficient adjusting the weights of the two embeddings of students, and $\mathbf{N} \in \mathbb{R}^{d \times n}$ is a trainable transformation matrix for the individual embeddings of students.

4.4.2 Model Optimization

Having the student and course representations constructed, we discuss how to generate the training data and learn the

student and course embeddings. We first separate the whole network by edge(grade) type, then given a view (grade type) r of the network, i.e., $\mathbf{G}_r = (\mathcal{C}, \mathcal{S}, \mathcal{E}_r, \mathcal{A})$, we use meta-path-based random walk[9] to generate node sequences. There are two meta-path schema in the student-course AMHEN, i.e., *student – course – student* or *course – student – course*. Finally, we apply a skip-gram [27, 28] over the node sequences to learn embeddings. The meta-path-based random walk strategy ensures that the semantic relationships between student nodes and course nodes with different grade types can be properly incorporated into the skip-gram model [9]. For a training pair (c_i, s_j) with grade type r , our objective is to maximize the probability:

$$P(s_j | c_i, r) = \frac{\exp(\mathbf{c}_{ir}^T \mathbf{s}_j')}{\sum_{s_k \in \mathcal{S}} \exp(\mathbf{c}_{ir}^T \mathbf{s}_k')} \quad (12)$$

where \mathbf{s}_k' is the context embedding of student node s_k . For a training pair (s_i, c_j) with grade type r , our objective is to maximize the probability:

$$P(c_j | s_i, r) = \frac{\exp(\mathbf{s}_i^T \mathbf{c}_{jr}')}{\sum_{c_k \in \mathcal{C}} \exp(\mathbf{s}_i^T \mathbf{c}_{kr}')} \quad (13)$$

where \mathbf{c}_{kr}' is the context embedding of course node c_k with grade type r . Finally, we use heterogeneous negative sampling to approximate the objective function $-\log P(s_j | c_i, r)$ for node pair (c_i, s_j) as

$$\text{loss}(c_i, s_j, r) = -\log \sigma(\mathbf{c}_{ir}^T \mathbf{s}_j') - \sum_{l=1}^L \mathbb{E}_{s_k \sim P(s_k)} [\log \sigma(-\mathbf{c}_{ir}^T \mathbf{s}_k')] \quad (14)$$

and the objective function $-\log P(c_j | s_i, r)$ for node pair (s_i, c_j) as:

$$\text{loss}(s_i, c_j, r) = -\log \sigma(\mathbf{s}_i^T \mathbf{c}'_{jr}) - \sum_{l=1}^L \mathbb{E}_{c_k \sim P(c_k)} [\log \sigma(-\mathbf{s}_i^T \mathbf{c}'_{kr})] \quad (15)$$

Here we define $P(s_k) = \frac{f(s_k)^{3/4}}{\sum_{i=1}^{|S|} f(s_i)^{3/4}}$ and $P(c_k) = \frac{f(c_k)^{3/4}}{\sum_{i=1}^{|C|} f(c_i)^{3/4}}$ according to the Skip-gram model[27], where f refers to the frequency of the node in each node type.

After optimizing the model with all the parameters learned, we reform the overall embedding for course i by concatenating its embeddings of all grade types.

$$\mathbf{c}_i = (\mathbf{c}_{i1}^T, \mathbf{c}_{i2}^T, \dots, \mathbf{c}_{im}^T)^T \quad (16)$$

5. TASKS

In this section, we describe five down-stream institutionally relevant tasks that can be performed by using the course representations constructed by the model approaches introduced in Section 4.

5.1 Course Similarity

An essential way to check the quality and fidelity of the course representations introduced in section 4 is to test whether they contain important features of courses that could differentiate between similar and dissimilar courses. To this end, an equivalency validation set of 1,351 course credit-equivalency pairs maintained by the Office of the Registrar were used for similarity based ground truth. A course is paired with another course in this set if a student can only receive credit for taking one of the courses at the university. For example, an honors and non-honors version of the same course will appear as a pair because faculty have deemed that there is too much overlapping material between the two for a student to receive credit for both.

To evaluate different course representations on the course equivalency validation set, we fixed the first course in each pair and ranked all the other courses according to their cosine similarity to the first course in descending order. We then noted the rank of the expected second course in the pair and describe the performance of each model on all validation pairs in terms of Mean Rank, Median Rank and Recall@10.

5.2 Enrollment Prediction

Enrollment prediction involves predicting the courses a student will enroll in, but not the grade they will receive. For this reason, it is considered a model of behavior, rather than an assessment model. The task could be potentially useful for the purpose of providing a normative course taking signal that could be used to provide a personalized sorting of course results (e.g., showing the courses a student is most likely to take that satisfy a remaining requirement) [32]. The input of the model in each time slice is a multi-hot vector representing the courses taken in the corresponding semester. However, the multi-hot representation has a large dimension of total number of courses and may not encode course features apparent in text descriptions of the course or graph-based methods. Therefore, we also evaluate substituting the multi-hot course input with the sum of pre-trained low-dimensional representations from other models, illustrated

in Figure 3. Performance on this task is reported in terms of Recall@10 and Mean Reciprocal Rank@10 (MRR@10). MRR evaluates recommender system models that produce a list of ranked items for queries. The reciprocal rank is the “multiplicative inverse” of the rank of the first correct item. MRR is defined as $\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^Q \frac{1}{\text{rank}_i}$, where rank_i represents the rank of the first correct recommended item for query i . For calculating MRR@10, the only difference is rank_i is reset to 0 if $\text{rank}_i > 10$.

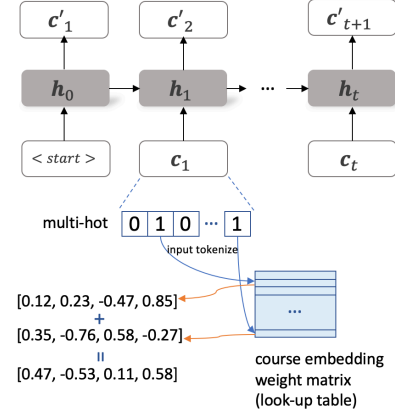


Figure 3: Illustration of the LSTM-based next-course prediction

5.3 Grade Prediction

Grade prediction is the basis for an assessment model that could aid adaptive sequencing of courses to achieve a particular goal. In previous work[21], a modified LSTM was designed to trace students’ course knowledge, which predicted students’ grades on enrolled courses in each semester. The model gives students the ability to choose their grade goal (A or B) or Pass/No-pass. A masked loss function was designed to enable the output to predict letter grade and Pass/No-pass independently. Two cut-offs (A or B) were also set to separate the letter grades into two levels (e.g., higher and lower than an ‘A’). The input of the LSTM grade prediction model is also a multi-hot vector with the position of grades students received for enrolled courses as 1 and other positions as 0. Because there are seven grade types for each course, the dimensions of the model input in each time slice is the number of courses multiplied by seven. As an alternative to the multi-hot input, we also evaluate the performance of the model using the course grade representations learned from the student-course AMHEN model in Section 4.4, which is illustrated in Figure 4, where \mathbf{g}_i represents the grades of courses taken in semester i and \mathbf{c}_i represents the courses taken in semester i . \mathbf{c}_{i+1} is concatenated with \mathbf{g}_i to incorporate the impact of the co-enrolling effect of courses in the predicted semester on grade prediction.

In addition, the student-course AMHEN model can also predict the grades of students by calculating the cosine similarities between student embeddings and course embeddings, and then predicting the grades by picking up the grade of each course that is most similar to the target student.

$$g(s_i, c_j) = \arg \max_r \cos(\mathbf{s}_i, \mathbf{c}_{jr}) \quad (17)$$

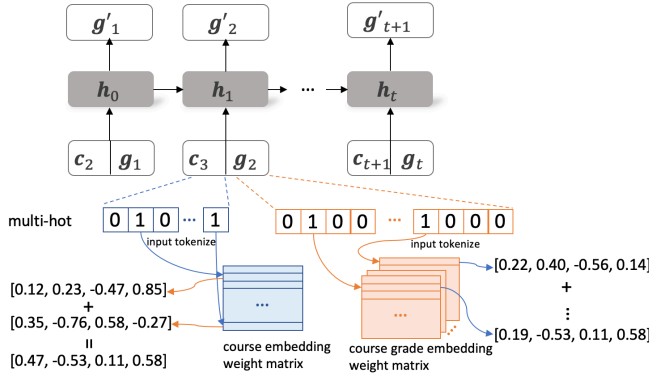


Figure 4: Illustration of the LSTM-based grade prediction

For the model without grade cut-off, there are seven grade types in the student-course AMHEN model representing A, B, C, D, F, Pass, and No-pass. A prediction is considered correct only if it is exactly the grade a student received in the data. For the models with grade cut-off (A or B), we group the letter grades not lower than the cut-off as a grade type, and the letter grades lower than the cut-off as another grade type in the student-course AMHEN model.

Both the enrollment prediction and grade prediction models were trained using a temporal train/test split, with Fall 2008 through Fall 2015 semesters serving as the training set and Spring 2016 as the testing semester.

5.4 Prerequisite prediction

Prerequisite course information is essential to encourage or mandate that students have the necessary foundational experience to be able to learn and succeed in the advanced stages of their degree. We used a set of 2,300 prerequisite course pairs, provided by the UC Berkeley Office of the Registrar, which contains 1,215 target courses, as a source of ground truth to test whether the grade prediction model encodes such prerequisite relationships between courses.

Prerequisite relationships between courses can be inferred by inferring an LSTM-based grade prediction model as described in [21] and illustrated in Figure 5. Note that, for this evaluation, only one time slice input of the binary-grade (A or lower than A) prediction trained LSTM is needed. We iterate over all the courses with only one-hot embedded in the ‘A’ position for that course, and feed the input, which is a concatenation of a target course and grade A of the input course, to the LSTM. During the iterations, the input course that boosted the probability of the ‘A’ position of target course to the largest ten values will be selected as candidate prerequisite courses for the target course. This approach is similar to the prerequisite skill inference conducted with DKT [35], but with a much larger vocabulary and with ground truth prerequisite structure to validate against. As with the other tasks, we also evaluate replacing the input of this model with representations from the student-course AMHEN graph-embedding approach.

A simple multinomial logistic regression can alternatively be

used to predict prerequisites courses using any arbitrary vector representation of a course. The input of the multinomial logistic regression during training is the vector representation of the target course, and the output is a multi-hot of the prerequisite courses for the target course. During testing, the output is a probability distribution across all courses where the most probable courses can be taken as the prerequisite predictions of the regression.

We classified all the models for the prerequisite course prediction task into two types, supervised and unsupervised, based on whether the model was learned using the official prerequisite course pairs. For the supervised models (i.e., using the regression), we applied 10-fold cross-validation to the 2,300 prerequisite course pairs. For the unsupervised models (i.e., LSTM-based inferences), described in Section 5.4, the LSTM with standard course multi-hots as input and with graph-based embeddings as input was trained first on the supervised task of predicting course grades, and was then inferred in an unsupervised manner (i.e., not using any prerequisite ground truth), to predict course prerequisites.

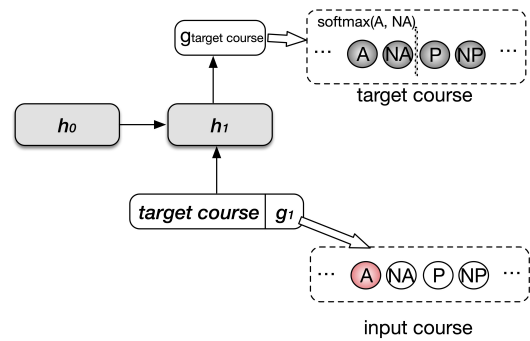


Figure 5: Prerequisite course prediction using LSTM-based grade prediction model[21]

5.5 Average Enrollment Prediction

Do the representations of courses created by various modeling techniques encode course popularity information? To answer this we test the course representations’ ability to predict the average enrollment size of each course. The data and models that perform well in this test may be indicative of the data and modeling paradigms that would work well for temporal versions of this model that could anticipate increases in course demand and allow institutions to better plan room and teaching staff allocations.

In order to check whether the different types of course embeddings encode information predictive of the number of enrollments, we use a simple a multi-layer perceptron to predict average enrollment per course using the different types of course embeddings introduced in section 4 as candidate inputs. RMSE is adopted as the error metric.

6. EXPERIMENT RESULTS

We begin this section by reporting a summary of only the best performing model and data source pairs used to construct the input representations for each of our five downstream model predictions tasks. This summarized set of best

Table 3: Evaluation of course representation models on various prediction tasks

Representation created by		Course similarity prediction		Enrollment prediction		Grade prediction	Prerequisite prediction		Avg-enroll-predict
Model	Data Source(s)	Mean/ Median Rank	Recall@10	Recall@10	MRR@10	Accuracy	Recall@10	Target	RMSE
bag-of-words	catalog	602/6 ^[33]	0.5370 ^[33]	0.3154	0.5216	-	0.5152	0.5938	42.4781
bag-of-words	syllabus	329/19	0.4270	0.3744	0.5103	-	0.5658	0.6352	48.8965
multi-c2v	enrollments, course meta-information	224/15 ^[33]	0.4485 ^[33]	0.3791	0.5576	-	0.6957	0.7733	42.4780
LSTM (multi-hot)	enrollments	584/58	0.2924	0.3967	0.5885	0.6952	0.3048 ^[21]	0.4486 ^[21]	51.4140
sc-AMHEN	enrollments, grades, catalog	288/11	0.4767	0.3882	0.5625	0.7008	0.7192	0.8000	52.3370

results are shown in Table 3. On the task of course similarity, a simple bag-of-words representation of the course catalog description performs best in terms of median rank and Recall @ 10 on our credit-equivalency pairs validation set. Enrollment histories provide the second best performing score using sc-AMHEN network-based embedding, followed by multi-c2v. Scoring similarly to multi-c2v was a simple BOW of the lms-syllabus data. On the task of predicting which courses a student will take next (enrollment prediction), an LSTM with a multi-hot input representation of courses taken in each semester provided the best performance in terms of both metrics. In this task, using pre-trained embeddings from the network-based or multi-c2v approach worked less well than multi-hot, followed by using the content-based representations as inputs, which performed worst. In grade prediction, the network-based method performed slightly better than the previous state-of-the-art LSTM. On the task of prerequisite prediction, the network-based approach performed best in recovering the ground-truth prerequisite relationships found in our institutional data. The multi-c2v approach was not far behind. The content-based and LSTM course representations did not perform nearly as well on this task. Finally, on the task of predicting the average enrollment of a course, multi-c2v provided the lowest RMSE, but with an almost identical score achieved by simple BOW of the course catalog description.

In the subsequent sections we provide a more detailed breakdown of performance of all model and data combinations on the tasks of course similarity, grade prediction, and prerequisite prediction. Results of enrollment prediction and average enrollment prediction are already shown in full in Table 3.

6.1 Course Similarity

The evaluation results on the equivalency validation set of 1,351 course credit-equivalency pairs are shown in Table 4. The bag-of-words representations (Tf-idf) generated from course catalog descriptions achieved better median rank and recall@10 than those generated from the course syllabus data. However, the mean rank of the catalog-based representations is the worst among all the models, which suggests there are many outliers where literal semantic similarity (bag-of-words) is very poor at identifying equivalent pairs. Concatenations of the bag-of-words based methods and course2vec-based method increased the evaluation met-

Table 4: Course similarity validation of all the course representations

Model	Mean/Median Rank	Recall @10
catalog	602/6	0.5372
syllabus	329/19	0.4270
course2vec (c2v)	244/21	0.3839
multi-c2v (mc2v)	224/15	0.4485
catalog+mc2v	132/3	0.6435
syllabus+mc2v	109/6	0.5798
catalog+syllabus+mc2v	79/3	0.6705
catalog+syllabus+mc2v (PCA dim: 300)	177/3	0.6544
LSTM	584/58	0.2924
sc-AMHEN(<i>u</i>)	288/11	0.4767
sc-AMHEN(<i>c</i>)	330/27	0.3603

rics, especially when the bag-of-words representations of catalog and syllabus were combined with the multi-factor course2vec representations, reaching a mean/median rank of 79/3 and recall@10 of 0.6705, the best among all the models. A Principal Component Analysis (PCA) transformation of the concatenated course vectors from 10,000 to 300 did not diminish the median rank metric, but slightly negatively affected average rank and recall. The course representations learned from the next-course prediction LSTM performed the worst among all the models. Course attribute embeddings sourced from the student-course AMHEN (sc-AMHEN) model, performed second best among all single representation models.

6.2 Grade Prediction

The *accuracy* of the grade predictions generated by the pure student-course AMHEN model (sc-AMHEN(*s, c*)), the LSTM model with multi-hot as input (LSTM(multi-hot)), and the LSTM model with course embeddings with different grade types (LSTM(*u, c*)) are listed in Table 5. Among the three models, the pure student-course AMHEN model is a kind of static model learned from students' enrollment data with grades and course catalog descriptions, while the two LSTM-based models are dynamic models taking into consideration not only the student enrollment data with grades, but also the sequential information (semester order) of the grades of enrolled courses. The grade prediction results show that the graph model, though static, could map the knowledge

Table 5: Grade prediction evaluation (accuracy)

Model	Type	Cut-off	Letter grade	Pass/No-pass	All
sc-AMHEN (\mathbf{s}, \mathbf{c})	static	-	0.5441	0.7972	0.5976
LSTM (multi-hot)	dynamic	-	0.6382	0.9079	0.6952
LSTM (\mathbf{u}, \mathbf{c})	dynamic	-	0.6418	0.9209	0.7008
sc-AMHEN (\mathbf{s}, \mathbf{c})	static	A	0.5526	0.7791	0.6004
LSTM (multi-hot)	dynamic	A	0.7523	0.8581	0.7633
LSTM (\mathbf{u}, \mathbf{c})	dynamic	A	0.7571	0.9135	0.7902
sc-AMHEN (\mathbf{s}, \mathbf{c})	static	B	0.8299	0.8205	0.8279
LSTM (multi-hot)	dynamic	B	0.8805	0.9178	0.8884
LSTM (\mathbf{u}, \mathbf{c})	dynamic	B	0.8817	0.9185	0.8895

levels of students on the features of courses with different grade types to a certain degree, resulting in prediction accuracies higher than 0.5 for all grade types and higher than 0.6 and 0.8 for binary grades (“not lower than cut-off” v.s. “lower than cut-off”, Pass v.s. No-pass) on average. Furthermore, the sequential information of students’ grades by semesters exhibited substantial importance as the prediction accuracy of the two LSTM-based models manifested superiority to the static student-course AMHEN model by a significant margin. Moreover, the course embeddings with different grade types learned from the student-course AMHEN model helped increase the accuracy of grade prediction over the multi-hot vectors as the input of the LSTM. The potential reasons could be the course embeddings with different grade types captured the knowledge relations among grades of a course and the relations among different courses, thus could represent the knowledge of students more accurately than multi-hot, which could not encode any knowledge relations among grades. Although the positive impact of incorporating grade embeddings on grade prediction (improvement at the 0.01 level) are not so salient as the advantage of bringing in sequential information (improvement at the 0.1 level), it is manifested in all the evaluations with different grade types.

6.3 Prerequisite prediction

The evaluation results of prerequisite course prediction are shown in Table 6. The supervised models performed dramati-

Table 6: Prerequisite course prediction

Model	Supervised	Pairs (Recall@10)	Target course
LSTM(one-hot)	X	0.3048	0.4486
LSTM(\mathbf{u}, \mathbf{c})	X	0.2423	0.3580
catalog	✓	0.5152	0.5938
syllabus	✓	0.5658	0.6352
mc2v	✓	0.6957	0.7733
sc-AMHEN(\mathbf{u}, \mathbf{c})	✓	0.7192	0.8000

cally better in reconstructing the prerequisite pairs. Among all types of course representations, the course embeddings and grade embeddings learned from the student-course AMHEN performed the best, reaching 71.92% of the prerequisite pairs

correctly predicted and 80% of all the target courses with at least one of their prerequisite course correctly predicted. For unsupervised models, we found one-hot representation of courses performed better than course and grade embeddings in the prerequisite course inference framework described in Section 5.4.

7. CONCLUSIONS

In this paper, we evaluated the utility of two content-sources of data about courses, catalog descriptions and syllabi, as well as enrollment histories and grades. We paired these sources with four different representations produced by simple bag-of-words, multifactor Course2vec, LSTM, and network-based embedding. We compared the performance of these pairings on five prediction tasks, course similarity, enrollment prediction, grade prediction, prerequisite prediction, and average enrollment prediction.

On the topic of the utility of syllabus data, which has not been evaluated before, we found that it showed benefit over catalog description data only in inferring prerequisite relationships (Recall of 0.5658 vs 0.5152), perhaps due to syllabi being the finer-grained source of content information about a course. In terms of course similarity signal, catalog description was markedly better than syllabus (Recall of 0.5372 vs 0.427) and our results indicate that catalog description, syllabus, and enrollment histories all bring some level of complementary information as the combination of all three performed better than any one or two combined. Enrollment data was used in the best scoring model in four of the five tasks, with only the best performing course similarity task model not utilizing enrollments. The nascent network-based approach performed well on all tasks, and was the top model in grade prediction and prerequisite prediction.

To conclude: (1) syllabus data is worth the effort to collect compared to catalog description for prerequisite prediction and (2) complements the catalog description and enrollment data on the course similarity task, (3) for prerequisite learning, supervised approaches based on embeddings perform much better than inferencing a pre-trained assessment model, (4) multifactor Course2vec often performs close to the more complex network-based approach on all tasks and (5) seeding the LSTM with course representations from the other models did not improve next-course prediction performance, while seeding with course grade representations from the student-course AMHEN model provided a small improvement in the grade prediction task.

8. LIMITATIONS AND FUTURE WORK

Our analyses were limited to data from a single large public institution in the US. Future work will need to evaluate multiple institutions of varying sizes, student demographics, and course taking policies in order to examine the generalizability of these approaches. In terms of models, we focused on simple text-based approaches and more complex neural models, both well established and nascent. Classical models of intermediary complexity were not evaluated.

We included tasks that have been common in EDM papers involving enrollment data; however, other institutional tasks exist that could be evaluated to produce an even more comprehensive analysis. These tasks include course preparation

recommendation [21, 20], degree or course attrition prediction, and future course demand forecasting.

Syllabi in their original form could be evaluated, instead of in bag-of-words form, in order to investigate if the positionality of words in the syllabi offered any additional predictive utility. Lastly, learning management system clickstream data, as well as content information in addition to the syllabus, could be leveraged to enhance both content-based and collaborative-based course representations. This combination of different modalities and scales of data is an identified open challenge for the field [14].

9. ACKNOWLEDGEMENTS

We would like to thank Sandeep Jayaprakash and Oliver Heyer from UC Berkeley’s Research Teaching, and Learning for supporting this research by facilitating the limited use of course syllabus data as pre-processed bag-of-words. We also thank the UC Berkeley Office of the Registrar for providing anonymized student enrollment data and access to course information APIs.

10. REFERENCES

- [1] G. Angus, R. D. Martinez, M. L. Stevens, and A. Paepcke. Via: Illuminating academic pathways at scale. In *Proceedings of the Sixth ACM Conference on Learning@ Scale*, pages 1–10, 2019.
- [2] C. Antunes. Acquiring background knowledge for intelligent tutoring systems. In *Proceedings of the 1st International Conference on Educational Data Mining*, 2008.
- [3] M. G. Brown, R. M. DeMonbrun, and S. D. Teasley. Conceptualizing co-enrollment: Accounting for student experiences across the curriculum. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, pages 305–309, 2018.
- [4] Y. Cen, X. Zou, J. Zhang, H. Yang, J. Zhou, and J. Tang. Representation learning for attributed multiplex heterogeneous network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1358–1368, 2019.
- [5] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang. Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 119–128, 2015.
- [6] W. Chen, A. S. Lan, D. Cao, C. Brinton, and M. Chiang. Behavioral analysis at scale: Learning course prerequisite structures from learner clickstreams. In *International Educational Data Mining Society*, 2018.
- [7] M. Dillon. Introduction to modern information retrieval: G. salton and m. mcgill. mcgraw-hill, new york (1983). 448 pp., isbn 0-07-054484-0, 1983.
- [8] M. Dong, R. Yu, and Z. A. Pardos. Design and deployment of a better course search tool: Inferring latent keywords from enrollment networks. In *European Conference on Technology Enhanced Learning*, pages 480–494. Springer, 2019.
- [9] Y. Dong, N. V. Chawla, and A. Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144, 2017.
- [10] A. Elbadrawy and G. Karypis. Domain-aware grade prediction and top-n course recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 183–190, 2016.
- [11] A. Elbadrawy and G. Karypis. Upm: Discovering course enrollment sequences associated with success. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 373–382, 2019.
- [12] A. Esteban, A. Zafra, and C. Romero. A hybrid multi-criteria approach using a genetic algorithm for recommending courses to university students. In *International Educational Data Mining Society*, 2018.
- [13] R. Farzan and P. Brusilovsky. Encouraging user participation in a course recommender system: An impact on user behavior. *Computers in Human Behavior*, 27(1):276–284, 2011.
- [14] C. Fischer, Z. A. Pardos, R. S. Baker, J. J. Williams, P. Smyth, R. Yu, S. Slater, R. Baker, and M. Warschauer. Mining big data in education: Affordances and challenges. *Review of Research in Education*, 44(1):130–160, 2020.
- [15] J. Gardner and C. Brooks. Coenrollment networks and their relationship to grades in undergraduate education. In *Proceedings of the 8th international conference on learning analytics and knowledge*, pages 295–304, 2018.
- [16] Q. Hu and H. Rangwala. Academic performance estimation with attention-based graph convolutional networks. In *Proceedings of The 12th International Conference on Educational Data Mining*, volume 69, page 78. ERIC, 2019.
- [17] L. Huang, C.-D. Wang, H.-Y. Chao, J.-H. Lai, and S. Y. Philip. A score prediction approach for optional course recommendation via cross-user-domain collaborative filtering. *IEEE Access*, 7:19550–19563, 2019.
- [18] S. M. Jayaprakash, E. W. Moody, E. J. Lauría, J. R. Regan, and J. D. Baron. Early alert of academically at-risk students: An open source analytics initiative. *Journal of Learning Analytics*, 1(1):6–47, 2014.
- [19] B. Jeon, E. Shafran, L. Breitfeller, J. Levin, and C. P. Rosé. Time-series insights into the process of passing or failing online university courses using neural-induced interpretable student states. In *Proceedings of the 12th International Conference on Educational Data Mining*, 2019.
- [20] W. Jiang and Z. A. Pardos. Time slice imputation for personalized goal-based recommendation in higher education. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 506–510, 2019.
- [21] W. Jiang, Z. A. Pardos, and Q. Wei. Goal-based course recommendation. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 36–45, 2019.
- [22] P. Kaur, A. Polyzou, and G. Karypis. Causal inference in higher education: Building better curriculums. In *Proceedings of the Sixth ACM Conference on Learning@ Scale*, pages 1–4, 2019.
- [23] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang,

- B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. *ICLR'17*, 2017.
- [24] W. Liu, P.-Y. Chen, S. Yeung, T. Suzumura, and L. Chen. Principled multilayer network embedding. In *2017 IEEE International Conference on Data Mining Workshops*, pages 134–141. IEEE, 2017.
- [25] Y. Luo and Z. A. Pardos. Diagnosing university student subject proficiency and predicting degree completion in vector space. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [26] H. Ma, X. Wang, J. Hou, and Y. Lu. Course recommendation based on semantic similarity analysis. In *2017 3rd IEEE International Conference on Control Science and Systems Engineering*, pages 638–641. IEEE, 2017.
- [27] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [29] R. Morsomme and S. V. Alferez. Content-based course recommender system for liberal arts education. In *Proceedings of The 12th International Conference on Educational Data Mining*, volume 748, page 753. ERIC, 2019.
- [30] S. Morsy and G. Karypis. Will this course increase or decrease your gpa? towards grade-aware course recommendation. *Journal of Educational Data Mining*, 11(2):20–46, 2019.
- [31] Z. A. Pardos, H. Chau, and H. Zhao. Data-assistive course-to-course articulation using machine translation. In *Proceedings of the Sixth Conference on Learning@ Scale*, pages 1–10, 2019.
- [32] Z. A. Pardos, Z. Fan, and W. Jiang. Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. *User Modeling and User-Adapted Interaction*, 29(2):487–525, 2019.
- [33] Z. A. Pardos and W. Jiang. Designing for serendipity in a university course recommendation system. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pages 350–359. ACM, 2020.
- [34] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.
- [35] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in neural information processing systems*, pages 505–513, 2015.
- [36] A. Polyzou, N. Athanasios, and G. Karypis. Scholars walk: A markov chain framework for course recommendation. In *Proceedings of the 12th International Conference on Educational Data Mining*, pages 396–401, 2019.
- [37] Z. Ren, X. Ning, A. Lan, and H. Rangwala. Grade prediction based on cumulative knowledge and co-taken courses. In *Proceedings of the 12th International Conference on Educational Data Mining*. ERIC, 2019.
- [38] Z. Ren, X. Ning, and H. Rangwala. Grade prediction with temporal course-wise influence. In *International Educational Data Mining Society*, 2017.
- [39] C. Romero, S. Ventura, P. G. Espejo, and C. Hervás. Data mining algorithms to classify students. In *Proceedings of the 1st International Conference on Educational Data Mining*, 2008.
- [40] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Chang. Network representation learning with rich text information. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Pick the Moment: Identifying Critical Pedagogical Decisions Using Long-Short Term Rewards

Song Ju, Guojing Zhou, Tiffany Barnes, Min Chi
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
{sju2, gzhou3, tmbarnes, mchi}@ncsu.edu

ABSTRACT

Identifying critical decisions is one of the most challenging decision-making problems in real-world applications. In this work, we propose a novel Reinforcement Learning (RL) based Long-Short Term Rewards (LSTR) framework for critical decisions identification. RL is a machine learning area concerning with inducing effective decision-making policies, following which result in the maximum cumulative *reward*. Many RL algorithms find the optimal policy via estimating the optimal *Q-values*, which specify the maximum cumulative reward the agent can receive. In our LSTR framework, the *long term* rewards are defined as *Q-values* and the *short term* rewards are determined by the *reward function*. Experiments on a synthetic GridWorld game and real-world Intelligent Tutoring System datasets show that the proposed LSTR framework indeed identifies the critical decisions in the sequences. Furthermore, our results show that carrying out the critical decisions alone is as effective as a fully-executed policy.

Keywords

Critical Decisions, Pedagogical Strategies, Critical Reinforcement Learning, Reinforcement Learning

1. INTRODUCTION

People make decisions every day, from minor decisions such as what to eat for lunch, to major decisions such as which college to enroll. This is equally true for tutorial interactions. Some decisions, such as what type of example to use may be minor, while others such as whether to give a new problem, or provide a solution for an old one, may not. In many cases the true significance of these decisions will not be known until well after the fact (much delayed), when students' exam scores come in or beyond. Moreover, for many such decisions, the significance is often individualized. So our research question is: *Given a long trajectory of decisions, can we automatically identify those which are critical to the outcome?*

Song Ju, Min Chi and Guojing Zhou "Pick the Moment: Identifying Critical Pedagogical Decisions Using Long-Short Term Rewards" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 126 - 136

Our work is primary concerned with identifying critical decisions in interactive learning environments such as Intelligent Tutoring Systems (ITSs) and educational games, where the human-agent interactions can be viewed as a temporal sequence of steps [2, 14]. Most ITSs are *tutor-driven* in that *the tutor* decides what to do next. For example, the tutor can *elicit* the subsequent step from the student either with prompting or without (e.g., in a free form entry window where each equation is a step). When a student enters an entry on a step, the ITS records its success or failure and may give feedback (e.g. correct/incorrect markings) and/or hints (suggestions for what to do next). Alternatively, the tutor can choose to *tell* them the next step directly. Each of such decisions affects the student's successive actions and performance and some may be more impactful than others. *Pedagogical policies* are used for the agent (tutor) to decide what action to take next in the face of alternatives.

Reinforcement Learning (RL) offers one of the most promising approaches to data-driven decision-making for improving student learning in ITSs. RL algorithms are designed to induce effective policies that determine the best action for an agent to take in any given situation so as to maximize a cumulative reward. In recent years, RL, especially Deep RL, has achieved superhuman performance in several complex games [25, 26, 3]. However, different from the classic game-play situations where the ultimate goal is to make the agent effective, in human-centric tasks such as ITSs, the ultimate goal is for the agent to make the *student-system interactions* productive and fruitful. A number of researchers have studied the application of existing RL algorithms to improve the effectiveness of ITSs [5, 24, 16, 21, 20, 19, 6, 27, 10, 31, 30, 32]. While promising, relatively little work has been done to analyze, interpret, explain, or generalize RL-induced policies. While traditional hypothesis-driven, cause-and-effect approaches offer clear conceptual and causal insights that can be evaluated and interpreted, RL-induced policies are often large, cumbersome, and difficult to understand. The space of possible policies is exponential in the number of domain features. It is therefore difficult to identify the system decisions that critical to desirable outcomes. This raises a major open question: *How can we identify the critical system interactive decisions that are linked to student learning?*

In this work, we propose Long-Short Term Rewards (LSTR) framework to identify *critical* decisions based on RL-induced policy. For RL-induced policies, we explore Deep Q-Networks (DQNs) [18] and also modify Deep Q-Networks based on

critical decisions referred as Critical DQN in the following. More specifically, we define critical decisions as those optimal decisions have to be made for the desired outcomes. To quantify their impacts, we define critical policy as the one which will carry out the optimal actions on the critical decisions while randomly on others. To identify critical decisions, we investigate on using an RL-induced policy's action-value functions (long term) alone and using both action-value functions (long term) and immediate rewards (short-term). The effectiveness of the proposed LSTR framework is evaluated on a synthetic GridWorld game and real-world Intelligent Tutoring System datasets. Our results show that the proposed LSTR framework indeed identifies critical decisions and moreover, carrying out the critical decisions alone is as effective as a fully-executed policy.

Our main contributions are summarized as follows: 1) we proposed the Long Short Term Rewards framework to identify critical decisions and evaluated on both a synthetic GridWorld game and real-world ITS dataset. 2) we proposed Critical DQN to improve the long term rewards in identifying critical decisions and investigated its advantages and disadvantages.

2. METHOD

We follow the conventional Reinforcement Learning (RL) notation. An agent interacts with an environment over a series of decision-making steps. The environment is framed as a Markov Decision Process (MDP). At each timestep t , the agent observes the state the environment is in, denoted s_t ; then the agent chooses an action from a discrete set of possible actions: $A \in (a_1, a_2, \dots, a_n)$. As a result, the environment provides a scalar *immediate reward* r . We assume that the future rewards are discounted by the factor $\gamma \in (0, 1]$, and the agent's goal is to maximize the expected discounted sum of future rewards, also known as the return. The return at time-step t is defined as $R_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'}$, where T is the last time-step in the episode.

The goal of the agent is to find the optimal action-value function $Q^*(s, a)$, which will result in the agent receiving the highest possible expected return, starting from state s , taking action a , and following the optimal policy π^* thereafter. Formally, we define the optimal action-value function as $Q^*(s, a) = \max_{\pi} \mathbb{E}[R_t | s_t = s, a_t = a, \pi]$. The optimal action-value function must follow the *Bellman Equation* shown in Equation 1, which states that the Q-value for a given state and action should be equal to the immediate reward obtained after taking that action, plus the discounted Q-value of the optimal action a' taken from the next state s' . Note that this is an expectation over the next states sampled from the environment.

$$Q^*(s, a) = \mathbb{E}_{s' \sim \mathcal{E}} [r + \gamma \max_{a'} Q^*(s', a') | s, a] \quad (1)$$

In our case, we follow the batch Reinforcement Learning formulation in that we have a fixed-size dataset \mathcal{D} consisting of all historical sample episodes and each episode is denoted as $s_1 \xrightarrow{a_1, r_1} s_2 \xrightarrow{a_2, r_2} s_3 \xrightarrow{a_3, r_3} \dots s_L$. To make this task more general, we assume that the state distribution and behavior policy that were used to collect this data are both unknown.

In the following, we will describe the two DRL algorithms explored: DQN and Critical DQN and two ways of defining critical decisions: long term reward vs. long-short term reward. Based on two types of DRL methods and two ways of identifying critical decisions, we will compare six different policies.

2.1 Two Types of Deep RL Policy

2.1.1 Original DQN

Deep Q-Network (DQN) is one of most promising approaches which is widely used on areas like robotics and video games [18]. Fundamentally, DQN is a version of Q-learning which uses neural networks to approximate the Q-values of the different state-action couples. In order to train the DQN algorithm, the two neural networks with equal architectures are employed: one for calculating the Q-value of the current state and action: $Q(s, a)$ and another neural network to calculate the Q-value of the next state and action: $Q(s', a')$. The former is the main network and its weights are denoted by θ and the latter is the target network, and its weights are denoted by θ^- . The *Bellman Equation* for DQN is shown in Equation 2 and it is trained through running a gradient descent algorithm to minimize the squared difference of the two sides of the equality.

$$Q(s, a; \theta) = \mathbb{E}_{s' \sim \mathcal{E}} [r + \gamma \max_{a'} Q(s', a'; \theta^-)] \quad (2)$$

The main network is trained on every training iteration, while the target network is frozen for a number of training iterations. Every k training iterations, the weights of the main neural network are copied into the target network. This is one of the techniques used in order to avoid divergence during the training process. In practice, DQN also uses an experience replay buffer to store the recently collected data and to uniformly sample (s, a, r, s') steps from it. By sampling uniformly, it breaks the correlations between samples of the same episode, making the learning process more robust and stable. In this work, as we are doing batch RL, our whole dataset will be the experience replay buffer, and it will not change during the training process.

Basically, DQN is a Q-learning method that it finds the optimal action-value function by updating its action-value function approximator recursively. Its major difference from the traditional RL is that a deep neural network is used as action-value function approximator and this allows it to deal with the tasks with high dimensional state space.

2.1.2 Critical DQN

In the original DQN, the Q functions are estimated based on the assumption that the optimal policy will be followed to the end. We define *critical policy* to be the one that the optimal decision will be carried out on critical decision points while random decisions on the rest. By not taking the optimal actions on non-critical decisions, we fundamentally change the dynamics of Bellman equation which assumed full-execution of the policy. Therefore we need to modify it so that it can incorporate our critical decisions into consideration.

For a single (s, a, r, s') tuple, the original Bellman Equation can be expressed as:

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a') \quad (3)$$

where r is the immediate reward for taking action a at state s ; γ is the discount factor; and $Q(s', a')$ is the action-value function for taking action a' at the subsequent state s' .

To induce a critical policy, we will modify the original Bellman equation based on whether a decision is critical or not. The intuition behind the Critical DQN is that if a decision is important, then the agent should take the best action otherwise the agent can randomly choose an action to take. Therefore, we have:

$$Q(s, a) = \begin{cases} r + \gamma \max Q(s', a') & s' \text{ is critical} \\ r + \gamma \text{mean} Q(s', a') & s' \text{ is non-critical} \end{cases} \quad (4)$$

In equation 4, to update the Q-value for any given s and a , it will consider whether the next state s' is critical or not. If it is critical, the maximum Q-value for s' will be used to update $Q(s, a)$; if the decision s' is non-critical, then the average Q-value among all the actions on s' will be used to update $Q(s, a)$.

Algorithm 1 presents the pseudo-code for critical DQN. First, it initializes all Q-values using the immediate rewards to avoid the bias of the neural network. In the main training loop, for each iteration, the algorithm first calculate the median threshold of Q-value difference over all the states. Then, for each (s, a, r, s') tuple, if the Q-value difference of s' is larger than the median threshold, we consider the decision on that state is critical and its value function is $\max_{a'} Q(s', a'; \theta^-)$; for non-critical decisions, their value function are defined as $\text{mean}_{a'} Q(s', a'; \theta^-)$. In this work, we assumes that half of the decisions in the training dataset are critical so that the median threshold is applied to separate critical and non-critical decisions quantitatively.

2.2 Two Types of Critical Rewards

2.2.1 Long Term Rewards (LongTRs)

In RL, $Q(s, a)$ is an estimation of the cumulative future rewards the agent will receive by taking action a at state s and following the policy to the end. If the Q-values for all the actions are the same, then it doesn't matter which action to take because all the actions will result in the same final reward. If the Q-value for one action is much larger than the others, then taking that action will have great impact on the future reward and this decision should be critical. So, the Long Term Reward is defined as how much cumulative future rewards the best action will obtain compared with the worst action. For this paper, we therefore define the Long Term Reward (LongTR) as:

$$\text{LongTR}(s) = \max_a Q(s, a) - \min_a Q(s, a) \quad (5)$$

which is the difference between the maximum and minimum Q-values in the state s . In general, the higher the LongTR, the more important the decision is.

Algorithm 1 Pseudocode of Critical DQN

```

1: Initialize the training dataset  $D$  as  $(s, a, r, s')$  tuples.
2: Initialize the Q function with random parameters  $\theta$ 
3: Initialize the target  $\hat{Q}$  function with parameters  $\theta^- = \theta$ 
4:
5: // Initialize  $Q(s, a)$  as immediate reward
6: for each  $(s_i, a_i, r_i, s'_i)$  in  $D$  do
7:   set  $y_i = r_i$ 
8: end for
9: Perform gradient descent on  $(y_i - Q(s_i, a_i; \theta))^2$ 
10: Reset  $\hat{Q} = Q$ 
11:
12: // Main Training Loop
13: for iteration  $k = 1, 2, \dots$  till convergence do
14:   Initialize empty array  $Q_{diffs}$ 
15:   for each  $(s_i, a_i, r_i, s'_i)$  in  $D$  do
16:      $Q_{diffs} \leftarrow (\max Q(s_i, a'; \theta^-) - \min Q(s_i, a'; \theta^-))$ 
17:   end for
18:    $\text{median\_threshold} = \text{median}(Q_{diffs})$ 
19:   for each  $(s_i, a_i, r_i, s'_i)$  in  $D$  do
20:     if terminal  $s'_i$  then
21:       Set  $y_i = r_i$ 
22:     else
23:        $Q_{diff} = \max Q(s'_i, a'; \theta^-) - \min Q(s'_i, a'; \theta^-)$ 
24:       if  $Q_{diff} > \text{median\_threshold}$  then
25:         Set  $y_i = r_i + \gamma \max_{a'} Q(s', a'; \theta^-)$ 
26:       else
27:         Set  $y_i = r_i + \gamma \text{mean}_{a'} Q(s', a'; \theta^-)$ 
28:       end if
29:     end if
30:   end for
31:   Perform gradient descent on  $(y_i - Q(s_i, a_i; \theta))^2$ 
32:   Every  $C$  steps reset  $\hat{Q} = Q$ 
33: end for

```

2.2.2 Long-Short Term Rewards (LSTRs)

For the LongTR, it only considers the cumulative future rewards but not immediate rewards. In a deterministic environment, LongTR is enough to identify critical decisions. But in a stochastic environment like the real world, some non-critical decision points would become critical and the LongTR can't detect their importance. For example, Figure 1 shows a simple MDP with seven states and one reward in the central state. Based on the LongTR, the decisions on S2 and S3 are critical because if doesn't move to the center, the agent will miss the +10 rewards. S1 is not critical based on LongTR because either move up or down doesn't affect collecting the reward as long as the agent takes the right action on state S2 or S3. However, in a stochastic environment, the agent should get the reward as soon as possible because the longer the path, the higher the risk to deorbit the rail. In RL, the LongTR can't learn the importance of state S1 but the immediate reward can. So, immediate rewards are served as Short Term Rewards (ShortTRs) in LSTR to complement the weakness of LongTRs that the agent should collect the rewards immediately without wandering.

2.3 Identifying & Evaluating Critical Decision

The effectiveness of our LSTR framework on identifying critical decisions is evaluated by the performance of **critical policy**. Unlike normal RL policies whose decisions are car-

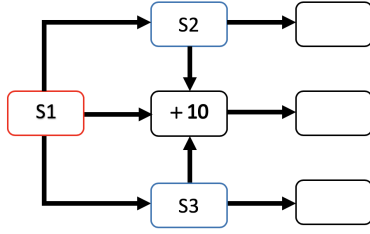


Figure 1: An MDP Example for LSTR

ried out all the time, our critical policy only follows the RL policy’s decisions at critical points and takes random actions otherwise. Ideally, the more accurate the critical decisions are identified, the better performance the critical policy should have.

More specifically, for a given training data set, we can induce RL policies following the original DQN or the Critical DQN, named as π or π_c , respectively. For each of the policy, there are two ways to identify critical decisions, using LongTR (L) and using LSTR (LS). Based on the rewards used for critical decision identification and the policy used for execution, we have the following six critical policies shown in the table 1.

Table 1: Six Critical Policies

	Critical Policy	Execution Policy	Rewards for Identifying Critical Decision
1	$\pi(L)$	π	LongTRs in π
2	$\pi(LS)$	π	LSTRs in π
3	$\pi_c(L_c)$	π_c	LongTRs in π_c
4	$\pi_c(LS_c)$	π_c	LSTRs in π_c
5	$\pi(L_c)$	π	LongTRs in π_c
6	$\pi(LS_c)$	π	LongTRs in π_c

The first four critical policies are a simple 2 (π vs. π_c) by 2 (L vs. LS) combination that each policy uses its own rewards to identify critical decisions. However, the connection between the critical decisions and the performance of critical policy is based on the assumption that the policy carried out at the critical points are optimal. In the Critical DQN, average Q-value is considered in the updating process and this may slow down the convergence. As a result, the policy π_c can be non-optimal. So, we include other two critical policies: $\pi(L_c)$ and $\pi(LS_c)$ which using the LongTR and LSTR from π_c to identify critical decisions but executing the policy π to make decisions. In general, the original DQN should converge faster and generate better policy than the Critical DQN.

3. SIMULATION ENVIRONMENT

3.1 GridWorld Description

The GridWorld environment is like a maze that the agent learns an optimal path from the start point to the end point. Figure 2 shows our GridWorld environment, which consists of 7 by 14 cells. The agent starts from the start state (right bottom corner), explores the 2D space and finishes at the end state (left upper corner). There are several walls in the GridWorld which are marked as black blocks. The agent state is simply represented by the X and Y coordinates.

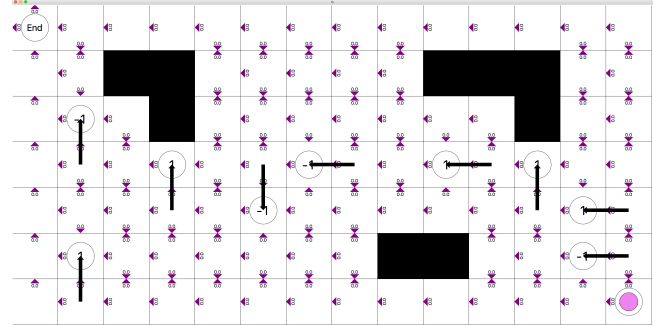


Figure 2: The Interface of the GridWorld Game

Action At each step, the agent can take three actions: up, down and left. In Figure 2, the possible actions for each state are labeled as small purple triangles that some states have three possible actions while some have two or one possible actions. The possible actions for each state are predefined in the environment so that the agent never hit the wall or the boundary.

Reward When moving in the GridWorld, there is -0.1 reward penalty for each step and the agent can collect -1 and +1 rewards. In order to simulate the real world, the reward function is designed in state-action-state way, $R(s, a, s')$. The black arrows indicate that only enter the reward state along with the arrow, the agent can get the reward -1 or +1. Otherwise, the agent won’t receive rewards. Furthermore, when the agent hits the reward state, it is forced to move left. This design aims to avoid the agent from collecting the same +1 reward repeatedly without forwarding to the terminal state.

Deterministic vs. Stochastic There are two transition settings in the GridWorld game: deterministic and stochastic. For deterministic transition setting, the next state is determined by the current state and action. For stochastic transition setting, the same state-action pair can result in different next states. For example, for deterministic setting, if the agent takes action ‘left’, then it will move to the left neighbor cell with 100 probability. For the stochastic transition setting, if the agent takes action ‘left’, then it only has 85% chance moving left, and 15% chance moving to other possible directions.

Finally, the performance of RL-induced policy in the GridWorld is evaluated by the final delayed reward which is the cumulative rewards during a trial. A good RL-induced policy should collect more +1 rewards, avoid -1 rewards and spend less steps to reach the goal.

3.2 Experiment Setup

Data Collection Since we focus on applying offline RL approaches to induce pedagogical policies, we induce all GridWorld policies offline. Following the data collection procedure in ITS, we collected the training data using a random policy. For the deterministic environment, we collected 500 randomly generated trajectories. Considering that the stochastic environment is more complicated, we collected

1000 trajectories for it.

Inferring Immediate Rewards Our LSTR framework requires immediate rewards to identify critical decisions, but our ITS data only have delayed rewards. Thus, we apply a Neural Network (NN) based approach to infer “immediate” rewards from delayed rewards. Given a trajectory to the NN as input, it outputs an “inferred” immediate reward for each step in the trajectory. The NN is trained using an additive error (the mean square error between the sum of inferred immediate rewards and the delayed rewards) as the loss function.

Critical Threshold Determination A key thing for identifying critical decisions is to choose an appropriate threshold on the long term and short term rewards that would not include too many trivial decisions but at the same would not exclude too many critical decisions. In order to pick a proper one, we conducted an analysis on the real and inferred immediate rewards and the Q-value difference. For immediate rewards, we would want to collect the large positive rewards and avoid the large negative rewards. Thus, rewards with a large absolute value should be considered as critical. Figure 3 shows the distribution of the real and inferred immediate rewards on the deterministic training dataset. The X axis shows the percentage of decisions in the dataset ranked by the value of the rewards (from large to small), and the Y axis shows value of the rewards. The threshold was set by allowing the real and critical rewards to identify similar numbers of critical decisions, which resulted in the value of 0.5. That is, if the ShortTR of a decision is greater than 0.5 or less than -0.5, it is critical. The same threshold was used for the stochastic dataset.

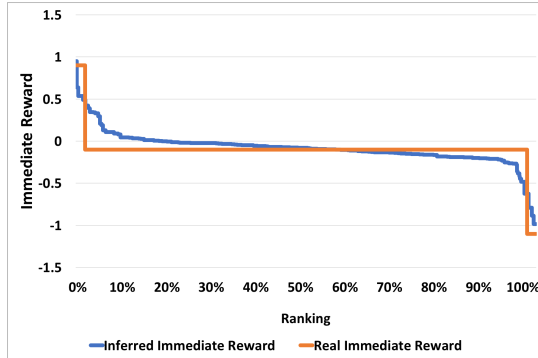


Figure 3: Immediate Reward Distribution

With the determined ShortTRs threshold, we explore different thresholds for the LongTRs in the experiment. The larger the threshold is (on percentage ranking), the more decisions will be carried out following the policy and the performance will in turn be better. To find a good balance between the number of critical decisions and the performance of the policy, we apply the policy with different thresholds (on percentage) at a 10% interval from 0% to 100% (0%, 10%, 20% ... 100%). Figure 4 shows an example distribution of the Q-value difference (LongTRs), calculated using the π policy in the deterministic dataset. For LSTRs, the critical decisions are the union from two set of critical decisions identified by LongTRs and ShortTRs separately.

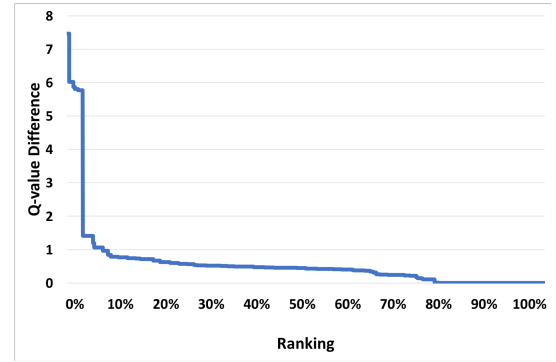


Figure 4: Q-value Difference Distribution

3.3 Results

We evaluate the performance of the six critical policies across two types of environment (deterministic vs. stochastic) with two types of immediate rewards (real vs. inferred). Figure 5 to Figure 8 shows the online evaluation results for the four possible settings. The X axis shows the percentage of decisions identified as critical ones based on the LongTRs on the training data. For example, 10% means the decisions with the top 10 percent LongTRs were considered as critical. The Y axis shows the cumulative rewards (the average of 100 trials under different random seeds) received by each critical policy. As expected, across all four figures, there is a general trend that the more decisions considered as critical, the better the policy will perform.

Overall, $\pi_c(LSc)$ and $\pi(LSc)$ outperforms the other four policies across all four settings. This suggests that when identifying critical decisions, LSTRs are more effective than LongTRs and Critical DQN is more effective than original DQN. This supported our expectation that both long term and short term rewards should be considered in critical decision identification and Critical DQN provides a better estimation of the long term rewards when the policies are partially carried out.

Next, we investigate in detail how the execution policy and the rewards (long term vs. long-short term) may impact the performance of the critical policies. More specifically, we present our results in five parts. First, compare the effectiveness of the original and Critical DQN on LongTRs. Second, investigate whether LSTRs can lead to better performance than LongTRs. Third, a mixed comparison between the critical decision recognition and policy execution. Fourth, exam the effectiveness of the inferred rewards. Finally, explore the performance of the Critical DQN with limited amount of training data.

3.3.1 Original DQN vs. Critical DQN on LongTRs

We first focus on comparing the original DQN policy $\pi(L)$ and the Critical DQN policy $\pi_c(L_c)$ with LongTRs, where the same policy was used for both execution and critical decisions identification. As we can see in all four figures, $\pi_c(L_c)$ outperformed $\pi(L)$ when no more than 50% decisions were considered as critical. More importantly, the fewer the critical decisions, the larger the gap is (except 0% which is totally random). This suggests that the Q-value difference in π_c is

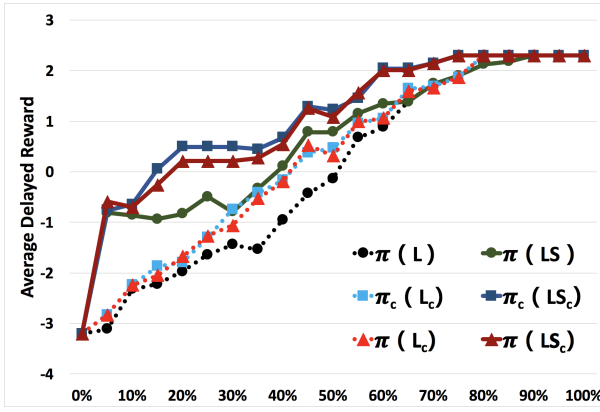


Figure 5: Deterministic GridWorld with Imm

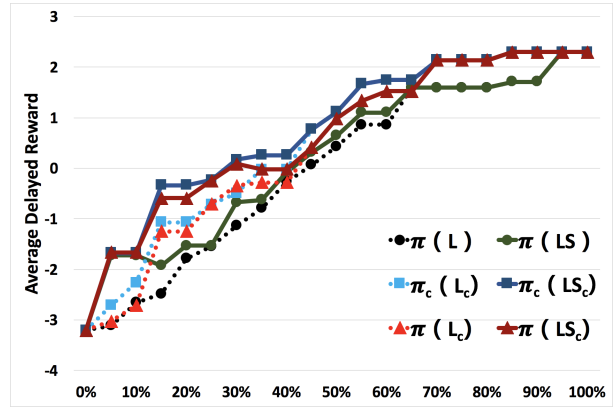


Figure 6: Deterministic GridWorld with Infer

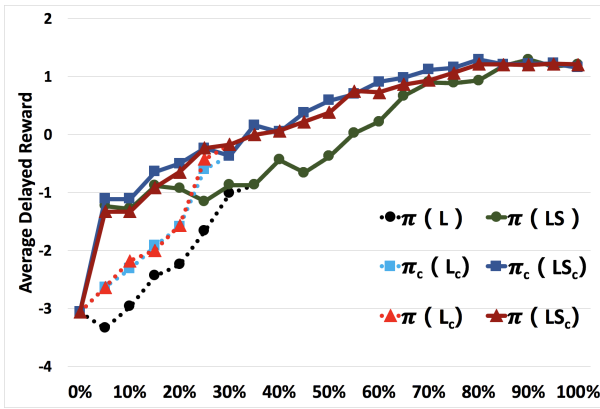


Figure 7: Stochastic GridWorld with Imm

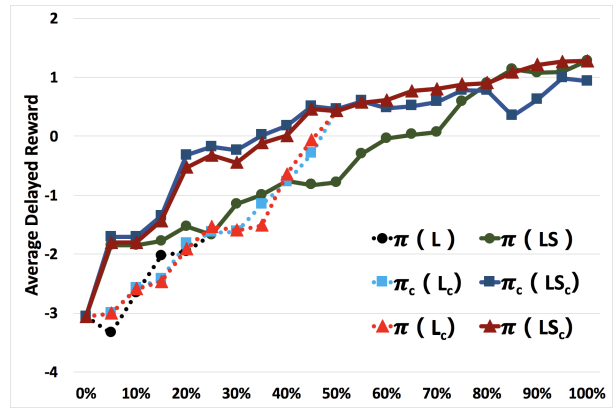


Figure 8: Stochastic GridWorld with Infer

more accurate and sensitive than that in π in identifying critical decisions. This result is not surprising because the Critical DQN already took the random execution of non-critical decisions into account in policy induction. Additionally, as expected, as the percentage of critical decisions increasing, both policies reached optimal. This suggests that our proposed Critical DQN could generate optimal policies as the DQN can do.

3.3.2 LSTRs vs. LongTRs

Second, we investigate how different rewards (LSTRs vs. LongTRs) may impact the performance of the policies. LSTR policies are shown in solid lines while LongTR policies are shown in dashed lines across Figure 5 to Figure 8. Here we focus on comparing the pair of policies with the same execution and critical decision identification policies such as $\pi(L)$ vs. $\pi(LS)$ and $\pi(Lc)$ vs. $\pi(LSc)$. Overall, results showed that the LSTR policies outperformed the LongTR policies when no more than 50% decisions were considered as critical (with few exceptions where the two policies have equal performance). More importantly, the performance of the LSTR policies had a sharp increase in the interval of 0% to 50% while the increase of the LongTR policies was relatively smooth. This resulted in a large gap between them when few decisions were considered as critical. This gap

gradually diminished as more decisions were included and disappeared eventually. This suggests that considering both the long term and short term rewards is more effective than considering the long term rewards only, especially when few decisions were considered as critical.

3.3.3 Mixed Comparison

There are two factors in the critical policy: execution policy and Rewards for critical decision identification. In this comparison, we fixed one factor and examined the impact of the other one on the critical policy. First, through fixing the execution policy as π , a comparison between Lc vs. L showed that the $\pi(Lc)$ outperformed the $\pi(L)$ across all the four Figures 5 to 8. Similar to this setting, we can get the same results that $\pi(LSc)$ is better than $\pi(LS)$. It means that the LSTRs in Critical DQN policy is more accurate to identify critical decisions than the original DQN. When fixing the Rewards for critical decision identification as Lc , a comparison between π vs. π_c showed that $\pi(Lc)$ and $\pi_c(Lc)$ have similar performance. This result also applies to $\pi(LSc)$ and $\pi_c(LSc)$. It indicates that the policy π and π_c could make similar decisions on critical points and the Critical DQN could induce optimal policy as the original DQN.

3.3.4 Inferred Rewards vs. Immediate Rewards

We also examined the effectiveness of the inferred immediate rewards by comparing them with real rewards. Figure 5 and 7 show the policies (immediate critical policy) induced using real immediate rewards; while Figure 6 and 8 show the policies (inferred critical policy) induced using inferred rewards. Through comparing the performance at 100%, all the inferred critical policies could reach the same optimal with the immediate critical policies in both deterministic and stochastic environments. This suggests that the inferred rewards could generate the optimal policy as real rewards. Then, the lines of inferred critical policies in Figure 6 and 8 have similar trend patterns with the lines in Figure 5 and 7, respectively. It means that the LSTRs calculated by inferred critical policies have similar distribution with the ones from immediate critical policies. In sum, the result indicates that inferred rewards could not only generate the optimal policy but also produce reliable LSTRs.

3.3.5 Data-Efficiency for Critical DQN

From the previous results, we could get a conclusion that when the policies π and π_c are both optimal, $\pi_c(L_c)$ is better than $\pi(L)$ regarding the identification of critical decisions. But what if we don't have enough data to train an optimal policy, how's the critical DQN performing?

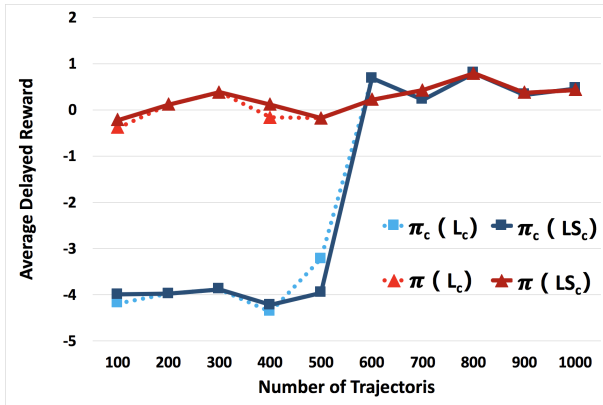


Figure 9: Original DQN vs. Critical DQN

Figure 9 shows the online performance of $\pi_c(L_c)$ vs. $\pi(L_c)$ and $\pi_c(LSc)$ vs. $\pi(LSc)$ as the number of training trajectories increasing. The X axis is the number of trajectories used to train the critical policies. The Y axis is the cumulative rewards (the average of 100 trails under different random seed) received by each critical policy. In this experiment, we applied the same rule to identify critical decision points for all the four policies and the only difference is which RL policy makes decision on the critical decision points. For $\pi(L_c)$, it means the critical decisions are identified by the LongTRs in π_c but execute π to make decisions in the online evaluation. It is the same for $\pi(LSc)$ that the LSTRs come from π_c while π decides what action to take. More specifically, the threshold for critical decisions is fixed by applying the same 0.5 threshold on short term rewards and 50% threshold on long term rewards.

The result shows that when the training dataset is less than 600 trajectories, the Critical DQN policies are worse than the original DQN policies. When the training dataset is

larger than 600 trajectories, they have similar performance. This suggests that the Critical DQN needs more data to converge to the optimal policy. But the LSTRs in Critical DQN is always good as the red lines $\pi(L_c)$ and $\pi(LSc)$ keep staying in the upper area from 100 to 1000 training trajectories. In summary, the Critical DQN could provide the best LSTRs to identify critical decisions but it needs more data to make good decision.

4. REAL-WORLD APPLICATION

4.1 Pyrenees Tutor Description

Pyrenees tutor is a web-based ITS for probability. It covers 10 major principles of probability, such as the Addition Theorem and Bayes' Rule. Pyrenees tutor provides step-by-step instruction and immediate feedback. Pyrenees tutor can also provide on-demand hints prompting the student with what they should do next. As with other systems, help in Pyrenees tutor is provided via a sequence of increasingly specific hints. The last hint in the sequence, the bottom-out hint, tells the student exactly what to do.

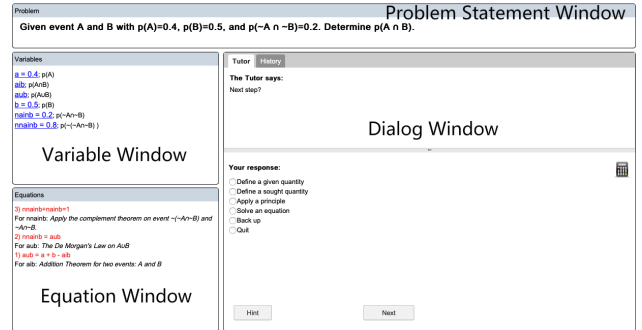


Figure 10: The Interface of the Pyrenees Tutor

Figure 10 shows the interface of Pyrenees, which consists of four windows. The top window shows the problem statement and doesn't change throughout the problem. In the dialog window, the upper part shows the instructions the tutor gives to the students such as an explanation of the current step or a prompt for the next step. At the same time, student enters an answer in the lower part of the dialog window such as selecting a choice or writing an equation. Any variables or equations generated through this process are shown on the left side of the screen for reference.

During tutoring, students are required to complete 4 phases: 1) pre-training, 2) pre-test, 3) training, and 4) post-test. In the pre-training phase, all students study the domain principles through a probability textbook by reviewing some examples and solving certain training problems. In the second phase, students take a pre-test which contains 14 problems. More specifically, the textbook is not available at this phase and students are not given feedback on their answers, nor are they allowed to go back to earlier questions. This is also true for the post-test. In phase 3, all students receive the same 12 rather complicated problems in the same order on Pyrenees tutor. Each of the 10 major principles needs to be applied at least twice in the training problems. For each problem, the average solving steps range from 20 to 50. Different from the pre- and post- test, students can access the

corresponding pre-training textbook and tutor help is available during this phase. Most importantly, the pedagogical policy works in this phase by deciding what action to take for each problem. In the training phase, each problem could have been provided as problem solving or worked example. Also, each step in the problem could have been provided as either a tell or elicit. Finally, all of the students complete a post-test with 20 problems. 14 of the problems are isomorphic to the pre-test given in phase 2. The remaining six are non-isomorphic complicated problems.

The performance of student learning is measured by the *normalized learning gain* (NLG) which is defined as $NLG = \frac{posttest - pretest}{1 - pretest}$ where 1 is the maximum score for both pre- and post- test. When grading the pre- and post- test, we use partial credit that each problem score is defined by the proportion of correct principle applications evident in the solution. For example, a student who correctly applied 4 of 5 possible principles would get a score of 0.8. All of the tests are graded in a double-blind manner by a single experienced grader. For comparison purposes, all test scores are normalized to the range of [0, 1].

4.2 Experiment Setup

Training Dataset Our training dataset contains a total of 1148 students' interaction log collected over six semesters' classroom studies (16 Fall to 19 Spring). The studies were assigned as a regular homework to students. During the studies, all students used the same tutor, followed the same general procedure, studied the same training materials, and worked through the same training problems.

From the student-system interaction logs, 142 features were extracted which describes the student learning state. All the 142 features can be categorized into five groups that **Autonomy** features describe the amount of work done by the student; **Temporal** features are the time related information during tutoring; **Problem Solving** features indicate the context of the problem itself; **Performance** features denote student's performance; and **Student Action** features record the student behavior information. For each problem, there are three possible actions: worked example (WE), problem solving (PS) and step decisions (SD). In WE, the student observes how the tutor solves a problem; in PS, the student solves the problem; in SD, student solves a portion of steps in a problem while the tutor shows how to solve the others. For reward, there's no immediate reward during tutoring and the delayed reward is the student's NLG.

Offline Learning and Evaluation The offline learning process follows the same process with the GridWorld in section 3.2. First, NN was applied to infer the immediate rewards for the training dataset. Then, critical policy π and π_c were induced based on the original DQN and the Critical DQN. Finally, we fixed the threshold of ShortTRs based on the elbows in the distribution and explored the relationship between different thresholds of LongTRs and the performance of the critical policies.

Different from the online evaluation in GridWorld game, we applied *off-policy* policy evaluation (OPE) metrics to evaluate the performance of the critical policies. In general, there are two types of OPE: model based and Importance Sam-

pling (IS) based. Song's work [12] showed that Per Decision Importance Sampling (PDIS) is the best metrics to evaluate the performance of RL-induced policies in the context of ITSs. So, PDIS was applied to evaluate the critical policies on the training dataset. More specifically, if a decision is identified as critical, the probability of taking that action is calculated by the softmax of Q-values among all the possible actions. On the contrary, if the decision is identified as non-critical, then the probability of taking that action is the random probability 1/3 as there are three possible actions for each problem.

4.3 Results

For Pyrenees tutor, we first present the offline evaluation results for all six critical policies. Then, we explore the identified critical decisions in the historical dataset.

4.3.1 Offline Evaluation Results

Figure 11 shows the offline evaluation results on Pyrenees tutor dataset. The X axis is the percentage of decisions identified as critical decisions in the historical dataset. The Y axis is the PDIS value. In general, the higher the PDIS, the better the policy.

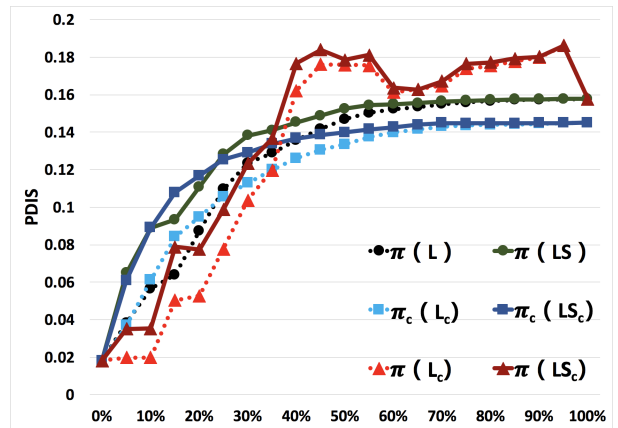


Figure 11: Offline Evaluation Results

First of all, the trend still holds that the more critical decisions, the better the policy would perform. When comparing within the dashed lines, there's no clear pattern before 40% threshold. However, $\pi(L_c)$ significantly outperformed the other two critical policies after 40%. The same trend occurs on the solid lines with LSTRs. The reason is that the Pyrenees dataset is not large enough for the Critical DQN to find an optimal policy, but the L_c and LS_c are still accurate to identify critical decisions. Furthermore, the performance jump around 50% demonstrates the reliability of the Critical DQN algorithm because in the pseudo-code 1, we already decide half of the decisions are critical decisions and the Figure 11 reflects this setting. As expected, the LSTR still outperforms LongTR that all the solid lines are above the corresponding dashed lines. In summary, the result reflects the effectiveness of LSTR in identifying critical decisions.

4.3.2 Exploring Critical Decisions

Table 2: Distribution of Critical Decisions in each Problem

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Long-Term Rewards	3%	16%	16%	13%	15%	11%	7%	7%	6%	6%
Short-Term Rewards	0%	1%	6%	6%	6%	10%	13%	19%	19%	20%
Long-Short Term Rewards	3%	14%	15%	12%	14%	11%	8%	8%	8%	8%

In order to further investigate the critical decisions identified by LSTRs in the tutor dataset, we analyzed where did they occur. 50% threshold for the LS_c in Figure 11 was applied to identify critical decisions in the tutor dataset and Table 2 shows the distribution of critical decisions in each problem. The first row represents the 10 problems in chronological order. The second row indicates the percentage of critical decisions identified by LongTRs in different problems. For example, 3% of critical decisions happens in P1 while 16% happens in P2. It indicates that the LongTRs focus on the critical decisions in the early to mid stage. In the meantime, the third row shows that the ShortTRs focus on the critical decisions in the late stage. The fourth row shows the critical decisions identified by LSTRs, which is the union set of the critical decisions from LongTRs and ShortTRs. Overall, critical decisions are evenly distributed among all the problems except the first one. It is not surprising that the first one is not so important because in the first problem, students are not familiar with the system and the policy needs more data to know the student status better. Furthermore, it reflects that the LongTRs and ShortTRs complement each other. If we only focus on LongTRs, we will miss the important decisions in the late stage, otherwise we will miss the important decisions in the early to mid stage.

5. RELATED WORK

5.1 RL For Pedagogical Policy Induction

Prior Research in Applying RL to Pedagogical Policy Induction can be roughly divided into classic RL vs. Deep RL approaches. The latter is highly motivated by the fact that the combination of deep learning (neural networks) and novel reinforcement learning algorithms has made solving complex problems possible in the last decade. For instance, the Deep Q-Network (DQN) algorithm [18] takes advantage of convolutional neural networks to learn to play Atari games observing the pixels directly. Since then, DRL has achieved success in various complex tasks such as the games of Go [25], Chess/Shogi [26], and robotic control [3]. One major challenge of these methods is *sample inefficiency* where RL policies need large sample sizes to learn optimal, generalizable policies. Batch RL, a sub-field of RL, aims to fix this problem by learning the optimal policy from a fixed set of a priori-known transition samples [15], thus efficiently learning from a potentially small amount of data and being able to generalize to unseen scenarios.

Prior research using classic RL approaches has applied both online and batch/offline approaches to induce pedagogical policies for ITSs. Beck et al. [4] applied temporal difference learning to induce pedagogical policies that would minimize the students’ time on task. Similarly, Iglesias et al. applied Q-learning to induce policies for efficient learning [10]. More recently, Rafferty et al. applied an online partially observable Markov decision process (POMDP) to induce policies for faster learning [19]. All of the models described above

were evaluated via simulations or classroom studies, yielding improved student learning and/or behaviors as compared to some baseline policies. Offline or batch RL approaches, on the other hand, “take advantage of previous collected samples, and generally provide robust convergence guarantees” [22]. Thus, the success of these approaches depends heavily on the quality of the training data. One common convention for collecting an exploratory corpus is to train students on ITSs using *random yet reasonable* policies. Shen et al. applied value iteration and least square policy iteration on a pre-collected exploratory corpus to induce a pedagogical policy that improved students’ learning performance [24, 23]. Chi et al. applied policy iteration to induce a pedagogical policy aimed at improving students’ learning gain [5]. Mandel et al. [16] applied an offline POMDP to induce a policy which aims to improve student performance in an educational game. All the models described above were evaluated in classroom studies and were found to yield certain improved student learning or performance relative to a baseline policy. Wang et al. applied an online DRL approach to induce a policy for adaptive narrative generation in educational game using simulations [29]; the resulting DRL-induced policies were evaluated via simulations only. In this work, based on the characteristics of our task domain, we focus on batch RL with neural networks, also known as batch Deep Reinforcement Learning (batch DRL) [11, 9].

5.2 Critical Decisions in Simulation

Student-Teacher framework is the most closely related work to our problem. In this framework, a “student” agent learns from the interaction with environment, while a “teacher” agent provides action suggestions to accelerate the learning process. Their research question is not what to advise but when to advise, especially with a limited budget of advice.

Clouse [7] was the first one studied the student-teacher framework in a student-initiated advising mode. They applied Q-value difference to measure the student’s confidence in a state and used it to decide when should the student ask for help. The results showed that compared with random asking, their approach could improve the learning speed significantly. Furthermore, the experiment demonstrated that not all the teacher’s advice are equally helpful. The same amount of advice can cause the student agent to take widely varying amounts of steps to find the optimal policy.

Torrey et al. [28] considered the student-teacher framework in teacher-initiated advising way. They considered an environment with a limited budget of advice and teacher decided when to give advice. They proposed several heuristic methods to determine when to give advice such as early advising, importance advising, mistake correcting and predictive advising. The results showed that mistake correcting has the best performance which indicates that advice can have the greatest impact when students make mistakes on important

states.

Zimmer et al. [33] modeled the when to advise problem as an RL problem. They learned a teaching policy with two actions: $A = \text{advise}, \text{noadvise}$ to decide when to give advice to the student. Compared with heuristic methods, the result showed that the teacher policy is effective because it can learn not only when to give advice, but also distinguish good and bad student agent that good agent chooses a lot of good actions and doesn't need advice while bad agent needs more.

Amir et al. [1] studied the jointly-initiated strategies for student-teacher learning framework. In their model, both student and teacher can initiate advising based on heuristic functions. The motivation of their work is to reduce the pressure of the teacher agent on monitoring the student constantly and make the framework more close to the real-life student-agent scenario. The result showed that the joint decision-making approach could reduce the attentions required from the teacher but still keep the student learning effectively.

Fachantidis et al. [8] explored the impact of advice quality in the student-teacher framework. They distinguished teacher agents to be an expert or a good teacher who provide optimal or sub-optimal advice. Also, a Q-teaching method was proposed to learn a teaching policy to decide when to give advice. Their results showed that the best performers are not always the best teachers and the Q-teaching approach is significantly more efficient than others.

In summary, prior works investigated the problem of when to give advice in simulated environments. They showed that Q-value difference is a robust and accurate heuristic function to estimate the importance of decision in interactive environments. However, prior works only considered RL-based student agent but not human students. In this work, we expand to a dataset of real-world ITS involving human students.

5.3 Exploiting Q-value Difference in ITSs

Some prior work exploited the Q-value difference between actions to simplify the decision-making process/problem in the context of ITS. For example Mitchell et al. relied on the Q-value difference to reduce the feature space [17]. More specifically, they proposed a policy evaluation metric, separation ratio for feature selection, which is defined as $\frac{2 * |Q(s, a_1) - Q(s, a_2)|}{(Q(s, a_1) + Q(s, a_2))}$ where $Q(s, a_i)$ is the Q value for the state-action pair (s, a_i) . The feature selection approach was then combined with RL to induce pedagogical policies for a dialog system, the Java tutor.

Zhou et al. [31] relied on Q-value difference to reduce the policy space. More specifically, they applied weighted decision tree with post-pruning to extract a compact set of 529 rules from a full set of 3706 rules. During the extraction, each rule was weighted by the Q-value difference between two alternative actions and thus increased the carry-out likelihood of more important decisions. The policies were empirically evaluated in a classroom study. Results showed that the full RL policy and the compact DT policy together were significantly more effective than a random policy and there is no significant difference between the full RL policy and

the compact DT policy.

Song et al. [13] proposed an ADRL framework to identify critical decisions and conducted an empirical study to test the effectiveness of the ADRL. In ADRL, two policies were induced that a positive policy aims to help student while a bad policy tries to hinder student learning. For a given state, if the two policies have different decisions and the Q-value difference is large enough, then this is a critical state and the decision is important. The results showed that critical phase exists in student learning that critical decisions always occur in groups and the more critical phase students have experienced, the better performance they have.

In sum, prior studies have considered the Q-value difference between actions as a heuristic function of action importance. The larger the difference, the more important the decision is. However, prior work didn't quantitatively study how large Q-value difference is a critical decision. In this work, we explored the Q-value difference thresholds by classifying decisions into two categories: critical and non-critical and evaluating the quality of the critical decisions.

6. CONCLUSIONS

In this study, we explored Long-Short Term Rewards to identify critical decisions in both synthetic Gridworld game and real-world ITS. Based on the LSTRs, we proposed Critical DQN to induce critical policy whose Q-value difference is more heuristic and sensitive to the decision importance. In order to investigate the effectiveness of LSTRs, we evaluated the performance of critical policies with different critical thresholds by online evaluation on GridWorld and offline evaluation on Pyrenees tutor's dataset. The results showed that the LongTRs from Critical DQN are significantly better than the original DQN. Through considering the ShortTRs, the LSTRs are significantly better than the LongTRs. However, the Critical DQN needs more data to converge to an optimal policy. In summary, through identifying critical decisions by the LSTRs, half of the decisions are trivial and carry out the optimal policy on the 50% decisions (critical ones) could achieve the similar effect of carrying out on all decisions.

In the future, we plan to generalize the LSTR framework to other domains in terms of interactive environments. Also, we hope to deploy a critical DQN policy on Pyrenees tutor and only carry out 50% decisions in a classroom study to test the critical decisions empirically.

7. ACKNOWLEDGEMENT

This research was supported by the NSF Grants: CAREER: Improving Adaptive Decision Making in Interactive Learning Environments(1651909), Integrated Data-driven Technologies for Individualized Instruction in STEM Learning Environments(1726550), and Generalizing Data-Driven Technologies to Improve Individualized STEM Instruction by Intelligent Tutors (2013502).

8. REFERENCES

- [1] O. Amir, E. Kamar, A. Kolobov, and B. J. Grosz. Interactive teaching strategies for agent training. *the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI*, pages 804–811, 2016.

- [2] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207, 1995.
- [3] M. Andrychowicz, B. Baker, et al. Learning dexterous in-hand manipulation. *arXiv preprint arXiv:1808.00177*, 2018.
- [4] J. Beck, B. P. Woolf, and C. R. Beal. Advisor: A machine learning architecture for intelligent tutor construction. *AAAI/IAAI*, 2000(552-557):1–2, 2000.
- [5] M. Chi, K. VanLehn, D. Litman, and P. Jordan. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction*, 21(1-2):137–180, 2011.
- [6] B. Clement, P.-Y. Oudeyer, and M. Lopes. A comparison of automatic teaching strategies for heterogeneous student populations. In *EDM*, 2016.
- [7] J. A. Clouse. On integrating apprentice learning and reinforcement learning. *PhD thesis*, 1996.
- [8] A. Fachantidis, M. E. Taylor, and I. P. Vlahavas. Learning to teach reinforcement learning agents. *Machine Learning and Knowledge Extraction*, 2017.
- [9] S. Fujimoto, E. Conti, M. Ghavamzadeh, and J. Pineau. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*, 2019.
- [10] A. Iglesias, P. Martínez, R. Aler, and F. Fernández. Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems. *Knowledge-Based Systems*, 22(4):266–270, 2009.
- [11] N. Jaques, A. Ghandeharioun, J. H. Shen, C. Ferguson, A. Lapedriza, N. Jones, S. Gu, and R. Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.
- [12] S. Ju, S. Shen, H. Azizzoltani, T. Barnes, and M. Chi. Importance sampling to identify empirically valid policies and their critical decisions. *EDM Workshop*, 2019.
- [13] S. Ju, G. Zhou, H. Azizzoltani, T. Barnes, and M. Chi. Identifying critical pedagogical decisions through adversarial deep reinforcement learning. *EDM Poster*, 2019.
- [14] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark. Intelligent tutoring goes to school in the big city. 1997.
- [15] S. Lange, T. Gabel, and M. Riedmiller. Batch reinforcement learning. In *Reinforcement learning*, pages 45–73. Springer, 2012.
- [16] T. Mandel, Y.-E. Liu, S. Levine, E. Brunskill, and Z. Popovic. Offline policy evaluation across representations with applications to educational games. In *AAMAS*, pages 1077–1084, 2014.
- [17] C. M. Mitchell, K. E. Boyer, and J. C. Lester. Evaluating state representations for reinforcement learning of turn-taking policies in tutorial dialogue christopher. *SIGDIAL*, pages 339–343, 2013.
- [18] V. Mnih, K. Kavukcuoglu, D. Silver, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.
- [19] A. N. Rafferty, E. Brunskill, T. L. Griffiths, and P. Shafto. Faster teaching via pomdp planning. *Cognitive science*, 40(6):1290–1332, 2016.
- [20] J. Rowe, B. Mott, and J. Lester. Optimizing player experience in interactive narrative planning: a modular reinforcement learning approach. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014.
- [21] J. P. Rowe and J. C. Lester. Improving student problem solving in narrative-centered learning environments: A modular reinforcement learning framework. In *AIED*, pages 419–428. Springer, 2015.
- [22] D. Schwab and S. Ray. Offline reinforcement learning with task hierarchies. *Machine Learning*, 106(9-10):1569–1598, 2017.
- [23] S. Shen and M. Chi. Aim low: Correlation-based feature selection for model-based reinforcement learning. In *EDM*, pages 507–512, 2016.
- [24] S. Shen and M. Chi. Reinforcement learning: the sooner the better, or the later the better? In *the 2016 Conference on User Modeling Adaptation and Personalization*, pages 37–44. ACM, 2016.
- [25] D. Silver, A. Huang, C. J. Maddison, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.
- [26] D. Silver, T. Hubert, J. Schrittwieser, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.
- [27] J. C. Stamper, M. Eagle, T. Barnes, and M. Croy. Experimental evaluation of automatic hint generation for a logic tutor. In *AIED*, pages 345–352. Springer, 2011.
- [28] L. Torrey and M. E. Taylor. Teaching on a budget: Agents advising agents in reinforcement learning. *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13*, pages 1053–1060, 2013.
- [29] P. Wang, J. Rowe, W. Min, B. Mott, and J. Lester. Interactive narrative personalization with deep reinforcement learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017.
- [30] G. Zhou, H. Azizzoltani, M. S. Ausin, T. Barnes, and M. Chi. Hierarchical reinforcement learning for pedagogical policy induction. In *Artificial Intelligence in Education - 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part I*, pages 544–556. Springer, 2019.
- [31] G. Zhou, J. Wang, C. Lynch, and M. Chi. Towards closing the loop: Bridging machine-induced pedagogical policies to learning theories. In *EDM*, 2017.
- [32] G. Zhou, X. Yang, H. Azizzoltani, T. Barnes, and M. Chi. Improving student-tutor interaction through data-driven explanation of hierarchical reinforcement induced pedagogical policies. In *UMAP*. ACM, 2020.
- [33] M. Zimmer, P. Viappiani, and P. Weng. Teacher-student framework: A reinforcement learning approach. *AAMAS Workshop Autonomous Robots and Multirobot Systems*, 2013.

Identifying At-Risk K-12 Students in Multimodal Online Environments: A Machine Learning Approach

Hang Li, Wenbiao Ding, Zitao Liu^{*}
TAL Education Group
16/F, Danling SOHO 6th Danling St
Haidian District Beijing, China, 100060
{lihang4, dingwenbiao, liuzitao}@100tal.com

ABSTRACT

With the rapid emergence of K-12 online learning platforms, a new era of education has been opened up. It is crucial to have a dropout warning framework to preemptively identify K-12 students who are at risk of dropping out of the online courses. Prior researchers have focused on predicting dropout in Massive Open Online Courses (MOOCs), which often deliver higher education, i.e., graduate level courses at top institutions. However, few studies have focused on developing a machine learning approach for students in K-12 online courses. In this paper, we develop a machine learning framework to conduct accurate at-risk student identification specialized in K-12 multimodal online environments. Our approach considers both online and offline factors around K-12 students and aims at solving the challenges of (1) multiple modalities, i.e., K-12 online environments involve interactions from different modalities such as video, voice, etc; (2) length variability, i.e., students with different lengths of learning history; (3) time sensitivity, i.e., the dropout likelihood is changing with time; and (4) data imbalance, i.e., only less than 20% of K-12 students will choose to drop out the class. We conduct a wide range of offline and online experiments to demonstrate the effectiveness of our approach. In our offline experiments, we show that our method improves the dropout prediction performance when compared to state-of-the-art baselines on a real-world educational dataset. In our online experiments, we test our approach on a third-party K-12 online tutoring platform for two months and the results show that more than 70% of dropout students are detected by the system.

Keywords

Dropout, retention, multimodal learning, online tutoring, K-12 education

1. INTRODUCTION

^{*}The corresponding author.

Hang Li, Wenbiao Ding, Songfan Yang and Zitao Liu "Identifying At-Risk K-12 Students in Multimodal Online Environments: A Machine Learning Approach" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 137 - 147

With the recent development of technologies such as digital video processing and live streaming, there has been a steady increase in the amount of K-12 students studying online courses worldwide. Online classes have become necessary complements to public school education in both developing and developed countries [31, 27, 37, 42, 35, 34, 36]. Different from public schools that focusing on teaching in traditional brick-and-mortar classrooms with 20 to 50 students, online classes open up a new era of education by incorporating more personalized and interactive experience [20, 33, 45, 9, 57].

In spite of the advantages of this new learning opportunity, a large group of online K-12 students fail to finish course programs with little supervision either from their parents or teachers. Students drop out of the class may be due to many reasons such as lack of interests or confidence, mismatches between course contents and students' leaning paths or even no immediate grade improvements from their parents' perspectives [37, 39, 25]. Therefore, it is crucial to build an early dropout warning system to identify such at-risk online K-12 students and provide timely interventions.

A large spectrum of approaches have been developed and successfully applied in predicting dropout in Massive Open Online Courses (MOOCs) [29, 47, 43, 58, 3, 5]. However, identifying dropout of K-12 students on online courses are significantly different from MOOCs based attrition prediction. The main differences are summarized as follows:

- **watching v.s. interaction:** Even though both learning are conducted in the online environment, learners' engagements on MOOCs and K-12 online platforms vary a lot [20]. In MOOCs, learners mainly watch the pre-recorded video clips and discuss questions and assignments with teaching assistants on the MOOC forums [18]. While in K-12 online courses, students frequently interact with the online tutors in a multimodal and immersive learning environment. The tutors may answer students' questions, summarize the knowledge points, take notes for students, etc.

- **spontaneous action v.s. paid service:** Learners on existing popular MOOC platforms such as Coursera¹, edX², etc. are adults, who aim at continuing

¹<https://www.coursera.org/>

²<https://www.edx.org/>

their lifelong learning in higher education and obtaining professional certificates such as Coursera’s Specializations and edX’s MicroMasters. MOOC learners are typically self-motivated and self-driven. On the contrary, most available K-12 online education choices are commercialized in service industry. Students pay to enroll online tutoring programs to strengthen their in-class knowledge levels and improve their grades in final exams. As a result, there are numerous out-of-class activities involved in K-12 online learning such as follow-ups from personal instructors, satisfaction survey and communications with students’ parents, etc. These out-of-class activities rarely appear in MOOC based learning.

- **high v.s. low dropout rate:** The dropout rate for MOOC based program is often as high as 70% - 93% [31, 52] while the dropout rate in K-12 online courses is below 20%.

Therefore, it is important to study approaches to identify at-risk K-12 online students and build an effective yet practical warning system. However, this task is rather challenging due to the following real-world characteristics:

- **multiple modalities:** K-12 online learning is conducted in an immersive and multimodal environment. Students and instructors interact with each other visually and vocally. There are a lot of multimodal factors that may influence the final decisions of dropout, ranging from interaction qualities between students and teachers, teaching speeds, volumes, emotions of the online tutors, etc.
- **length variability:** Students join and leave the online platforms independently, which results in a collection of observation sequences with different lengths. A dropout prediction system should be able to (1) make predictions for students with various lengths of learning histories; and (2) handle newly enrolled students.
- **data imbalance:** The overall dropout rate for K-12 online classes is usually below 20%, which makes the training samples particularly imbalanced.

The objective of this work is to study and develop models that can be used for accurately identifying at-risk K-12 students in multimodal online environments. More specifically, we are interested in developing models and methods that can predict risk scores (dropout probabilities) given the history of past observations of students. We develop a data augmentation technique to alleviate class imbalance issues when considering the multi-step ahead prediction tasks. We conduct extensive sets of experiments to examine every component of our approach to fully evaluate the dropout prediction performance.

Overall this paper makes the following contributions:

- We design various types of features to fully capture both in-class multimodal interactions and out-of-class

activities. We create a data augmentation strategy to simulate the time-sensitive changes of dropout likelihood in real scenarios and alleviate the data imbalance problem.

- We design a set of comprehensive experiments to understand prediction accuracy and performance impact of different components and settings from both qualitative and quantitative perspectives by using a real-world educational dataset.
- We push our approach into a real production environment to demonstrate the effectiveness of our proposed dropout early warning system.

The remainder of the paper is organized as follows: Section 2 discusses the related research work of dropout prediction in both public school settings and MOOCs scenarios. Comparisons with relevant researches are discussed. In Section 3, we introduce assumptions when building a practical at-risk student identification system and formulate the prediction task. Section 4, we describe the details about our prediction framework, which include (1) extracting various types of features from both online classroom recordings and offline activity logs (See Section 4.1); and (2) data augmentation technique that helps us create sufficient training pairs and overcomes the class imbalance problem (See Section 4.2). In Section 5, we (1) quantitatively show that our model supports better dropout predictions than alternative approaches on an educational data derived from a third party K-12 online learning platform and (2) demonstrate the effectiveness of our proposed approach in the a real production environment. We summarize our work and outline potential future extensions in Section 6.

2. RELATED WORK

Dropout prediction and at-risk student identification have been gaining popularity in both the educational research and the AI communities. Understanding the reasons behind dropouts and building early warning systems have attracted a growing interest of academics in the learning analytics area. Broadly speaking, existing research regarding dropout prediction can be categorized by learning scenarios and divided into two categories: (1) public school dropout (See Section 2.1); and (2) MOOCs dropout (See Section 2.2).

2.1 Public School Dropout

Education institutions are faced with the challenges of low student retention rates and high number of dropouts [45, 32]. For examples, in the United States, almost one-third of public high school students fail to graduate from high school each year [40, 7] and over 41% of undergraduate students at four-year institutions failed to graduate within six years in Fall 2009 [38]. Hence, research work has focused on predicting the dropout problem and developing dropout prevention strategies [40, 41, 8, 55, 13, 30, 10, 49]. Zhang and Rangwala develop an at-risk student identification approach based on iterative logistic regression that utilizes all the information from historical data from previous cohorts [59]. The state of Wisconsin creates a predictive dropout early warning system for students in grades six through nine and provides predictions on the likelihood of graduation for over

225,000 students [30]. The system utilizes ensemble learning and is built on the steps of searching through candidate models, selecting some subsets of best models, and averaging those models into a single predictive model. Lee and Chung address the class imbalance issue using the synthetic minority over-sampling techniques on 165,715 high school students from the National Education Information System in South Korea [33]. Ameri et al. consider different groups of variables such as family background, financial, college enrollment and semester-wise credits and develop a survival analysis framework for early prediction of student dropout using Cox proportional hazards model [1].

2.2 MOOCs Dropout

With the recent boom in educational technologies and resources both in industry and academia, MOOCs have rapidly moved into a place of prominence in the mind of the public and have attracted a lot of research attentions from many communities in different domains. Among all the MOOC related research questions, dropout prediction problem emerges due to the surprisingly high attrition rate [54, 19, 26, 23, 44, 56, 6, 11, 12, 20]. Ramesh et al. treat students' engagement types as latent variables and use probabilistic soft logic to model the complex interactions of students' behavioral, linguistic and social cues [43]. Sharkey et al. conduct a series of experiments to analyze the effects of different types of features and choices of prediction models [47]. Kim et al. study the in-video dropouts and interaction peaks, which can be explained by five identified student activity patterns [25]. He et al. propose two transfer learning based logistic regression algorithms to balance the prediction accuracy and inter-week smoothness [21]. Tang et al. formulate the dropout prediction as a time series forecasting problem and use a recurrent neural network with long short-term memory cells to model the sequential information among features [50]. Both Yang et al. and Mendez et al. conduct survival analysis to investigate the social and behavioral factors that affect dropout along the way during participating in MOOCs [58, 39]. Detailed literature surveys on MOOC based dropout prediction are reviewed comprehensively in [51, 5].

In this work, we focus on identifying at-risk students in K-12 online classes, which is significantly distinguished from dropout predictions in either public school or MOOCs based scenarios. In the K-12 multimodal learning environment, the learning paradigm focuses on interactions instead of watching. The interactions come from different modalities, which rarely happen in traditional public schools and MOOC based programs of higher education. Furthermore, as a paid service, K-12 online learning involves both in-class and out-of-class activities and both of them contain multiple factors that could lead to class dropouts. These differences make existing research works inapplicable in K-12 online learning scenarios. To the best of our knowledge, this is the first research that comprehensively studies the dropout prediction problem in K-12 online learning environments from real-world perspectives.

3. PROBLEM FORMULATION

3.1 Assumptions

In order to characterize the K-12 online learning scenarios, we need to carefully consider every cases in the real-world

environment and make reasonable assumptions. Without loss of generality, we have the following assumptions in the rest of the paper.

ASSUMPTION 1 (RECENCY EFFECT). *Time spans between the date of dropout and the date of last online courses vary a lot. Students may choose to drop the class right after one course or quit after two weeks of no course. Therefore, the per-day likelihood of dropout should be time-aware and the closer to the dropout date, the more accurate the dropout prediction should be.*

ASSUMPTION 2 (MULTI-STEP AHEAD FORECAST). *The real-world dropout prediction framework should be able to flexibly support multi-step ahead predictions, i.e., the next-day and next-week probabilities of dropout.*

3.2 The Prediction Problem

In this work, our objective is to predict the value of future status for the *target student* given his or her past learning history, i.e., observations collected from K-12 online platforms. More specifically, let \mathbf{S} be the collection of all students and for each student $s, s \in \mathbf{S}$, we assume that we have observed a sequence of n_s past observation-time pairs $\{< \mathbf{x}_j^s, t_j^s > \}_{j=1}^{n_s}$, $\mathbf{x}_j^s \in \mathbf{X}^s$, and $t_j^s \in \mathbf{T}^s$, such that $0 < t_j^s < t_{j+1}^s$, and \mathbf{x}_j^s is the observation vector made at time (t_j^s) for student s . \mathbf{X}^s and \mathbf{T}^s represent the collections of observations and timestamps for student s . Correspondingly, let \mathbf{Y}^s be the collection of indicators of status (*dropout, ongoing or completion*) of student s at each timestamp, i.e., $\mathbf{Y}^s = \{y_j^s\}_{j=1}^{n_s}$. Let Δ be the future time span in multi-step ahead prediction. Time $t_{n_s+\Delta}^s$ ($\Delta > 0$) is the time at which we would like to predict the student's future status $\hat{y}_{t_{n_s+\Delta}^s}^s$.

Please note that we omit the explicit student index s in the following sections for notational brevity and our approach can be generalized into a large samples of student data without modifications.

4. THE PREDICTION FRAMEWORK

The dropout prediction for K-12 online courses is a time-variant task. A student who just had the class should have a smaller dropout probability compared with a student haven't take any class for two weeks. Therefore, when designing a real applicable approach of dropout prediction, such recency effect, i.e., Assumption 1, has to be considered. In this work, we extract both static and time-variant features from different categories to capture the factors leading to dropout events comprehensively (See Section 4.1). Furthermore, we create a label augmentation technique that not only alleviates the class imbalance problem when building predictive framework for K-12 online classes, but incorporates the recency effect into label constructions (See Section 4.2). The learning of our dropout model is discussed in Section 4.3 and the overall learning procedure is summarized in Section 4.4.

4.1 Features

In this section, we develop a distinguished set of features for at-risk student identification from the real-world K-12 online learning scenarios, which can be divided into three categories: (1) in-class features that focus on K-12 students'

online behaviors during the class (See Section 4.1.1); (2) out-of-class features that consider as much as possible real-world factors happened after the class, which may influence the dropout decisions (See Section 4.1.2); and (3) time-variant features that include both historical performance of teachers and aggregated features of student activities within fixed-size windows (See Section 4.1.3).

4.1.1 In-class Features

Different from adults who continue their learning in higher education on MOOC based platforms, K-12 students come for grade improvements. This intrinsic difference in their learning goals leads to contrasting learning behaviors. Adult learners of MOOCs study independently by various activities, such as viewing lecture videos, posting questions on MOOC forums, etc. This results in various types of in-class click-stream data, which are shown to be effective in dropout prediction in many existing research works [15, 51, 54, 6, 12, 11, 20]. However, such click based activities barely happen in K-12 online scenarios. Instead, there are frequent voice based interactions between K-12 students and their teachers. The teachers not only make every effort to clarify unsolved questions that students remain from their public schools, but are responsible for arousing students' learning interests and building their studying habits. Therefore, we focus on extracting in-class multimodal features specializing in K-12 tutoring scenarios from the online classroom videos. We categorize our features as follows. Table 1 illustrates some examples of in-class features from different categories.

- **Prosodic features:** speech-related features such as signal energy, loudness, Mel-frequency cepstral coefficients (MFCC), etc.
- **Linguistic features:** language-related features such as statistics of part-of-speech tags, the number of interregnum words, distribution of length of sentences, voice speed of each sentence, etc.
- **Interaction features:** features such as the number of teacher-student Q&A rounds, the numbers of times teachers remind students to take notes etc.

To extract all the features listed in Table 1, we first extract audio tracks from classroom recordings on both teacher's and student's sides. Then we extract acoustic features directly from classroom audio tracks by utilizing the widely used open-sourced tool, i.e., *OpenSmile*³. We obtain classroom transcriptions by passing audio files to a self-trained automatic speech recognition (ASR) module. After that, we extract both linguistic and interaction features from the conversational transcripts. Finally, we concatenate all features from above categories and apply a linear PCA to get the final dense in-class features. The entire in-class feature extraction workflow of our approach is illustrated in Figure 1.

Please note that due to the benefits of online steaming, both students' and teachers' videos are recorded separately and hence, there is no voice overlap in the video recordings. This

³<https://www.audeering.com/opensmile/>

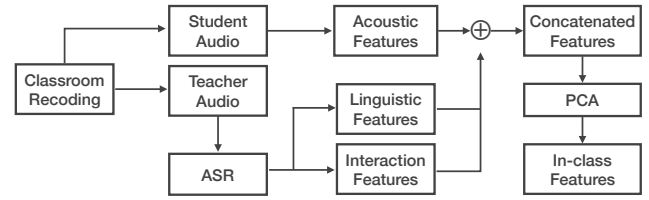


Figure 1: The workflow of our in-class features extraction. ASR is short for automatic speech recognition.

avoids the unsolved challenge of speaker diarization [2]. Similar to Blanchard et al. [4], we find that publicly available ASR service may yield inferior performance in the noisy and dynamic classroom environments. Therefore, we train our own ASR models on the classroom specific dataset based on a deep feed-forward sequential memory network, proposed by Zhang et al. [60]. Our ASR has a word error rate of 28.08% in our classroom settings.

4.1.2 Out-of-class Features

As we discussed in Section 1, personalized K-12 online tutoring is a paid service in most countries. Besides the course quality itself, there are multiple other factors in such service industry that may change customers' minds to drop the class. Therefore, out-of-class features play an extremely important role in identifying at-risk students in real-world K-12 online scenarios, which are typically ignored in previous literatures. In this work, we collect and summarize all the available out-of-class features and divide them into the following two categories. The illustrative examples are listed in Table 1.

- **Pre-class features:** Pre-class features capture the students' (or even their parents') behaviors before taking the class, such as purchasing behaviors, promotion negotiations, etc. Examples: the number of rounds of conversation and negotiation before the class, how much the discount student received, etc.
- **Post-class features:** Post-class features model the offline activities in such paid K-12 online services. For examples, students and their parents receive follow-ups based on their previous class performance and give their satisfaction feedbacks. Another example is that students may request changes to their course schedules.

4.1.3 Time-variant Features

Besides in-class and out-of-class features, we manually design time-variant features to model the changes of likelihood of students' dropout intentions. Cases like a student just had a class compared to a student had a class two weeks ago should be explicitly distinguished when constructing features. Therefore, we create time-variant features by utilizing a lookback window approach on students' observation sequences. More specifically, for a given timestamp, we only focus on previously observed activities of each student within a period of time. The length of lookback windows varies from 1 to 30 days. Sufficient statistics are extracted as time-variant features from each lookback window. Meanwhile, we

Table 1: List of examples in in-class, out-of-class, and time-variant features.

Category	Type	Examples
In-class	Prosodic	the average signal energies of student and teacher the average loudness of student and teacher the Mel-frequency cepstral coefficients of audio tracks from student and teacher the zero-crossing rates of student and teacher ...
	Linguistic	# of sentences per class of student and teacher # of pause words per class of student and teacher average lengths of sentences per class of student and teacher voice speeds (char per second) of student and teacher ...
	Interaction	# of teacher-student Q&A rounds # of times the teacher reminds the student to take notes and summarization # of times the teacher asks the student to repeat # of times the teacher clarifies the student's questions ...
Out-of-class	Pre-class	# of days since the student places the online course order # of courses in the student's order # of conversations between the sales staff and the students (or their parents) the discount ratio of the student's order ...
	Post-class	# of follow-ups after the student took the first class # of words in the latest follow-up report # of times the student reschedules the class the follow-up ratio, i.e., # of follow-ups divided by # of taken courses ...
Time-variant	Historical performance	# of courses taught by each individual teacher in total # of courses the student had in total historical dropout rates historical average time span between classes ...
	Lookback window	# of courses taken in past one/two/three weeks # of courses the student scheduled in past one/two/three weeks # of positive/negative follow-up reports in past one/two/three weeks the average time span of classes taken in past month ...

compute historical performance features to reflect the teaching experience and performance for each individual teacher. Table 1 shows some examples of time-variant feature we use in our dropout prediction framework.

- **Lookback window features:** The lookback window features aggregate important statistics from students' observations within a fixed-length lookback window, such as the numbers of courses taken in past one, two, three weeks.
- **Historical performance features:** The historical features aggregate each teacher's past teaching performance, which represent the overall teaching quality profiles. They include total numbers of courses and students taught, historical dropout rates, etc.

4.2 Data Augmentation

According to Assumptions 1 and 2 and the problem formulation in Section 3.2, a real-world early warning system is

supposed to flexibly support multi-step ahead predictions for each student, i.e., given any future time span Δ , the system computes the probability of student's status $\hat{y}_{t_{n_s}+\Delta}^s$. The predicted probability should be able to dynamically adapt when the values of Δ get changed. The multi-step ahead assumption essentially requires the approach to make predictions at a more fine-grained granularity of $\langle \text{student}, \text{timestamp} \rangle$ pair, i.e., $\langle s, t_{n_s+\Delta}^s \rangle$, instead of student level, i.e., s . This poses a challenging question: due to the fact that only about 20% of K-12 students drop their online classes, *how do we tackle the class imbalance problem when extracting $\langle \text{student}, \text{timestamp} \rangle$ training pairs from a collection of multimodal observation sequences (either completion or dropout) in K-12 online scenarios?*

Let \mathbf{S}_1 and \mathbf{S}_2 be the set of student indices of dropout and non-dropout students, i.e., $\mathbf{S}_1 = \{i | y_{n_i}^i = \text{dropout}\}$, and $\mathbf{S}_2 = \{j | y_{n_j}^j = \text{completion}\}$. Let \mathbf{P} and \mathbf{N} be the sets of positive (dropout) and negative (non-dropout) $\langle \text{student}, \text{timestamp} \rangle$ pairs. By definition, \mathbf{P} and \mathbf{N} are constructed as follows:

$$\begin{aligned}
\mathbf{P} &= \{ \langle \mathbf{x}_{n_i}^i, t_{n_i}^i \rangle \mid i \in \mathbf{S}_1 \} \\
\mathbf{N} &= \{ \langle \mathbf{x}_k^i, t_k^i \rangle \mid i \in \mathbf{S}_1, k \in \mathbf{T}_i \setminus t_{n_i}^i \} \\
&\cup \{ \langle \mathbf{x}_k^j, t_k^j \rangle \mid j \in \mathbf{S}_2, k \in \mathbf{T}_j \}
\end{aligned} \quad (1)$$

Similar to many researches such as fraud detection [53], the sizes of \mathbf{P} and \mathbf{N} are typically very imbalanced. While in some cases the class imbalance problem may be alleviated by applying an over-sampling algorithm on the minority class sample set, the diversity of the available instances is often limited. Therefore, in this work, we propose a time-aware data augmentation technique that artificially generates pseudo positive (dropout) $\langle \text{student}, \text{timestamp} \rangle$ pairs.

More specifically, for each dropout student i in \mathbf{S}_1 , we set a lookback window with length Λ where $\Lambda \leq t_{n_i}^i - t_{n_i-1}^i$. For each timestamp \tilde{t}_l^i in the lookback window such that

$$\max(t_{n_i-1}^i, t_{n_i}^i - \Lambda) < \tilde{t}_l^i < t_{n_i}^i. \quad (2)$$

We generate its corresponding pseudo positive training pair $\langle \tilde{\mathbf{x}}_l^i, \tilde{t}_l^i \rangle$ as follows: $\tilde{\mathbf{x}}_l^i = \mathcal{F}(\mathbf{X}^s, \mathbf{T}^s)$ where $\mathcal{F}(\cdot, \cdot)$ is the generation function. The choices of $\mathcal{F}(\cdot, \cdot)$ are flexible and vary among different types of features (See Section 4.1). In this work, for in-class and out-of-class features, we aggregate all the available features till \tilde{t}_l^i and re-compute the time-variant features according to timestamp \tilde{t}_l^i . We use $\tilde{\mathbf{P}}$ to represent the collection of all positive training pairs generated from dropout students in \mathbf{S}_1 . Figure 2 illustrates how the pseudo positive training pairs are generated.

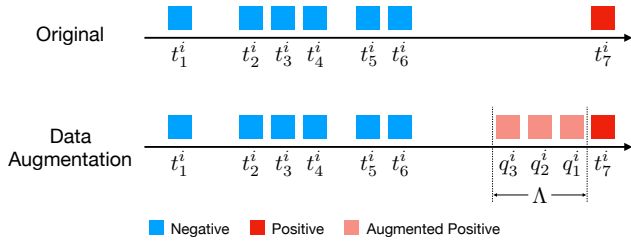


Figure 2: Graphical illustration of the data augmentation technique.

Besides, we assign a time-aware weight to each pseudo positive training pair to reflect the recency effect in Assumption 1. For each pseudo pair $\langle \tilde{\mathbf{x}}_l^i, \tilde{t}_l^i \rangle$, the corresponding weight w_l^i is computed by

$$w_l^i = \mathcal{G}\left(\frac{t_{n_i}^i - \tilde{t}_l^i}{\Lambda}\right) \quad (3)$$

where the weighting function $\mathcal{G}(\cdot)$ takes the normalized time span between each timestamp of pseudo pair and the exact dropout date as input and outputs a normalized weighting score to reflect our confidence on the “positiveness” of the simulated training pairs. The closer to the dropout date, the

larger the confidence weights should be. The choices of $\mathcal{G}(\cdot)$ are open to any function that gives response values ranging from 0 to 1, such as linear, convex or concave functions illustrated in Figure 3.

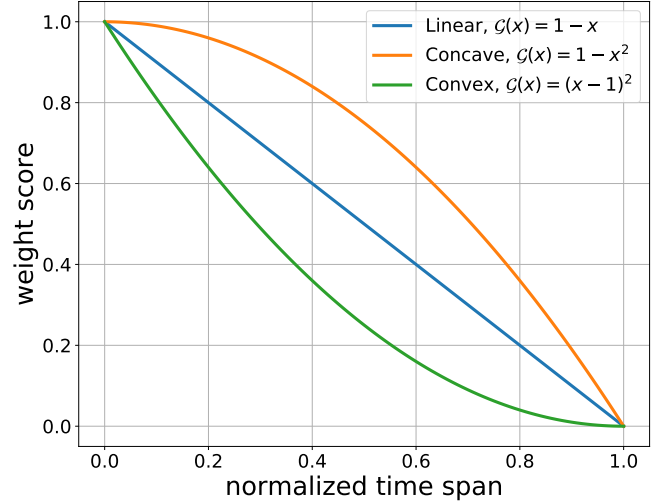


Figure 3: Graphical illustration of different weighting function of $\mathcal{G}(\cdot)$.

The effect of different choices of weighting function is discussed in Section 5.4. The augmented training set $\tilde{\mathbf{P}}$ and the corresponding time-aware weights are used in the model training in Section 4.3.

4.3 Model Learning

In the learning stage, we combine the original training set (\mathbf{P} and \mathbf{N}) with the augmented set $\tilde{\mathbf{P}}$ for model training. Even though the data augmentation alleviates the class imbalance problem, i.e., improving the positive example ratio from 0.1% to 10%, the imbalance problem still exists. Therefore, we employ the classical weighted over-sampling algorithm on positive pairs to further reduce the imbalance effect. Here, the weights of the original positive examples in \mathbf{P} are set to 1 and pseudo positive examples’ weights are computed by $\mathcal{G}(\cdot)$ in Section 4.2. Here, since the dropout datasets are usually small compared to other Internet scaled datasets, we choose to use Gradient Boosting Decision Tree⁴ (GBDT) [16] as our prediction model. The GBDT exhibits its robust predictive performance in many well studied problems [24, 48].

4.4 Summary

The overall model learning procedure of our K-12 online dropout prediction can be summarized in Algorithm 1.

5. EXPERIMENTAL EVALUATION

In this section, we will (1) introduce our dataset that is collected from a real-world K-12 online learning platform and the details of our experimental settings (Section 5.1); (2) show that our approach is able to improve the predictive performance when compared to a wide range of classic

⁴<https://scikit-learn.org/stable/modules/ensemble.html#gradient-tree-boosting>

Algorithm 1 Model learning procedure of the K-12 online dropout prediction.

INPUT:

- A set of K-12 students \mathbf{S} and their corresponding multimodal classroom recordings and activities logs.
- The length of lookback window Λ .
- The choice of weighting function $\mathcal{G}(\cdot)$.

PROCEDURE:

- 1: // Feature extraction
- 2: Extract in-class features from multimodal recordings, see Section 4.1.1.
- 3: Extract out-of-class features from student activities logs, see Section 4.1.2.
- 4: Extract time-variant features, see Section 4.1.3.
- 5: Concatenate three types of features above.
- 6: // Label generation and augmentation
- 7: Create original positive and negative training pair sets, i.e., \mathbf{P} and \mathbf{N} , see eq.(1).
- 8: Generate the augmented pseudo positive training sets, i.e., $\tilde{\mathbf{P}}$ and the corresponding weights, see eq.(3).
- 9: // Model learning
- 10: Conduct weighted over-sampling on the union of \mathbf{P} and $\tilde{\mathbf{P}}$.
- 11: Train the GBDT model on the over-sampled positive examples and original negative examples.

OUTPUT:

- The GBDT dropout prediction model Ω .
-

baselines (Section 5.2); (3) evaluate the impacts of different sizes of lookback windows, different weighting functions in data augmentation and feature combinations (Section 5.3, Section 5.4 and Section 5.5); and (4) deploy our model into the real production system to demonstrate its effectiveness (Section 5.6).

We would also like to note that hyper parameters used in our methods are selected (in all experiments) by the internal cross validation approach while optimizing models' predictive performances. In the following experiment, we set the size of lookback window to 7 and the impact of window size is discussed in section 5.4. We choose to use the convex weighting function when conducting pseudo positive data augmentation.

5.1 Experimental Setting

5.1.1 Data

To evaluate the effectiveness of our proposed framework, we conduct several experiments on a real-world K-12 online course dataset from a third-party online education platform. We select 3922 registered middle school and high school students from August 2018 and February 2019 as our samples. All the features listed in Section 4.1 are computed and extracted from daily activity logs on the platform. In our dataset, 634 students choose to drop the class and the dropout rate is 16.16%. The average time span of the students on the platform is about 86 days, which provide us 338428 observational $\langle \text{student}, \text{time stamp} \rangle$ sample pairs in total. We randomly select 80% of students and use their corresponding $\langle \text{student}, \text{time stamp} \rangle$ sample pairs as training set and the remaining 20% of students' sample pairs for testing propose. The data augmentation technique discussed in Section 4.2 is only applied in training set.

5.1.2 Multi-step Ahead Prediction Setting

To fully examine the dropout prediction performance, we evaluate the model's predictions in terms of different multi-step ahead time spans, i.e, given a current timestamp, we predict the outcome (dropout or non-dropout) in the next X days, where X ranges from 1, 2, \dots , 14.

5.1.3 Evaluation Metric

Similar to [18, 50, 17, 15, 51, 21], we evaluate and compare the performance of the different methods by using the Area Under Curve (AUC) score, which is the area under the Receive Operating Characteristic curve (ROC) [14]. An ROC curve is a graphic plot created by plotting the true positive rate (TPR) against the false positive rate (FPR). In our dropout prediction scenario, the TPR is the fraction of the "at-risk" predicted students who truly drop out. The FPR is the ratio of the falsely predicted "dropout" students to the true ones. The AUC score is invariant to data imbalance issue and it does not require additional parameters, for models comparisons. AUC score reaches its best value at 1 and the worst at value 0.

5.1.4 Baselines

We compare our proposed approach with the following representative baseline methods: (1) Logistic Regression (LR) [28], (2) Decision Tree (DT) [46] and (3) Random Forest (RF) [22]. LR, DT and RF are all trained on the same set of features defined in Section 4.1 with our proposed method. The training set is created by using eq.(1).

5.2 The Overall Prediction Performance

The results of these models are shown in Figure 4. As we can see from the Figure 4, we have the following observations:

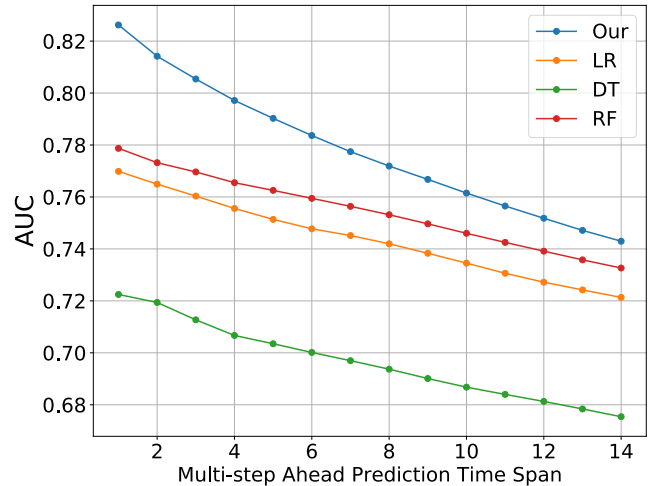


Figure 4: The overall prediction performance with different multi-step ahead time spans in terms of AUC scores.

- First, our model outperforms all other methods in terms of AUC scores on different future time spans, which demonstrates the effectiveness of our approaches with positive data augmentation. By adding more diverse pseudo positive training pairs with the corresponding decaying confidence weights, the GBDT model is able to learn the dropout patterns from multiple factors.

- Second, as we increase the lengths of time spans of multi-step ahead prediction, all the models' performances decrease accordingly. Our approach achieves AUC score of 0.8262 in the task of next day prediction while the performance downgrades to 0.7430 in the next two-week prediction task. We believe this is because of the truth negative mistakes the models make, i.e., the model thinks the students will continue but they drop classes in next two weeks. This indicates that without knowing more information from the students, the ML models have very limited ability in predicting the long-term outcomes of student status, which also reflects the fact that there are many factors that could lead to the dropouts.
- Third, comparing LR, DT, and RF, we can see, the DT achieves the worst performance. This is because of its instability. With small number of training data, the DT approach suffers from fractional data turbulence. The RF approach remedies such shortcomings by replacing a single decision tree with a random forest of decision trees and the performance is boosted. Meanwhile, as a linear model, the LR is not powerful enough to accurately capture the dropout cases.

5.3 Impact of Sizes of Lookback Windows

As we can see, the number of augmented positive training pairs is directly determined by the size of lookback window Λ . Therefore, to comprehensively understand the performance of our proposed approach, we conduct experimental comparisons on different sizes of lookback windows. We vary the window size from 3, 7, and 14. Meanwhile, we add a baseline with no data augmentation. The results are shown in Figure 5.

From Figure 5, we can see that the size of lookback windows has a positive relationship on AUC scores with the length of time span in multi-step ahead prediction. When conducting short-term dropout predictions, models trained on data augmentation with smaller size of lookback window outperform others. As we gradually increase the time span of future predictions, the more the model looks back, the higher the prediction AUC score it achieves. Overall, the model trained with 7-day lookback window has the best performance across different multi-step ahead time spans in terms of AUC scores.

5.4 Impact of Different Weighting Functions

In this section, we examine the performance changes by varying the forms of weighting functions. More specifically, we compare the prediction results of using the convex function to results of the other choices. The results are shown in Figure 6. As we can see from Figure 6, the convex option outperforms other choices by a large margin across all different multi-step ahead time spans. When computing the over-sampling weights of pseudo training examples, the convex function gives more weights to the most recent examples, i.e., examples close to the timestamp of true dropout observations. This also confirms the necessity of considering the recency effect assumption (Assumption 1) when building the dropout prediction framework.

5.5 Impact of Different Features

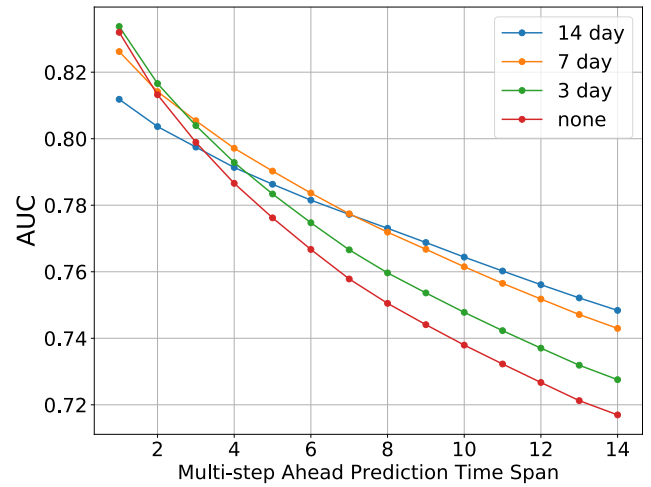


Figure 5: Models trained on data augmented by different size of lookback windows with different multi-step ahead time spans in terms of AUC scores. *none* represents the model training without any lookback data augmentation.

In this subsection, we systematically examine the effect of different types of features by constructing following model variants:

- In: only the in-class features are used.
- Out: only the out-of-class features are used.
- Time: only the time-variant features are used.
- In+Time: it eliminates the contribution of *Out* features and only uses features from *In* and *Time*.
- Out+Time: it eliminates the contribution of *In* features and only uses features from *Out* and *Time*.
- In+Out: it eliminates the contribution of *Time* features and only uses features from *In* and *Out*.
- In+Out+Time: it uses the combination of all the features from *In*, *Out* and *Time*.

Meanwhile, we also consider different multi-step ahead prediction settings, i.e., next 7-day prediction and next 14-day prediction and the prediction results are shown in Table 2. From Table 2, we observe that (1) by considering all three types of features individually, the model trained from *Out* features yields the best performance. Moreover, when comparing *In*, *Out* to *In+Time*, *Out+Time*, we obtain the significant performance improvement by adding *Out* features. These indicate the fact that dropout prediction for K-12 online scenarios are very different from MOOC based dropout prediction. The out-of-class activities and the quality of the service play an extremely important role in the prediction task; and (2) by utilizing all the sets of features, we could be able to achieve the best results in both prediction tasks.

Table 2: Experimental results of different types of features and different lengths of multi-step ahead time span in terms of AUC scores.

	In	Out	Time	In+Time	Out+Time	In+Out	In+Out+Time
Multi-step ahead time span - 7 day	0.6249	0.7764	0.6992	0.7145	0.7759	0.7768	0.7774
Multi-step ahead time span - 14 day	0.6251	0.7386	0.6766	0.6932	0.7393	0.7420	0.7430

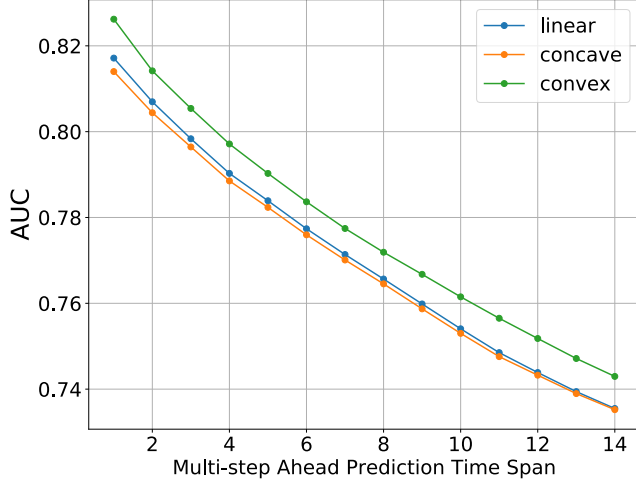


Figure 6: Models trained on data augmented by different choices of weighting functions with different multi-step ahead time spans in terms of AUC scores.

5.6 Online Performance

We deployed our at-risk student warning system in the real production environment on a third-party platform between February 2nd, 2019 to April 1st, 2019. To watch the system performance in practice, we conduct the next-day prediction task where the system predicts the dropout probability for each on-going student at 6 am in the morning. All the students are ranked by their dropout probabilities and the top 30% of students with highest probabilities are marked as at-risk students. At the end of each day, we obtain the real outcome of all the students who drop the class. We conduct the overlap comparison between the predicted top at-risk students (30% of total students) and the daily dropouts and we are able to achieve that more than 70% of dropout students are detected by the system.

6. CONCLUSION

In this paper, we present an effective at-risk student identification framework for K-12 online classes. Compared to the existing dropout prediction researches, our approach considers and focuses on the challenging factors such as multiple modalities, length variability, time sensitivity, class imbalance problems when learning from real-world K-12 educational data. Our offline experimental results show that our approach outperforms other state-of-the-art prediction approaches in terms of AUC scores. Furthermore, we deploy our model into a production environment and we are able to achieve that more than 70% of dropout students are detected by the system. In the future, we plan to explore the opportunity of using deep neural networks to capture heterogeneous information in the K-12 online scenarios to enhance

the existing prediction pipeline.

7. ACKNOWLEDGMENTS

The authors would like to thank Nan Jiang from Peking University and Lyu Qing from Tsinghua University, for their contributions to this research during their internship in TAL Education Group.

8. REFERENCES

- [1] S. Ameri, M. J. Fard, R. B. Chinnam, and C. K. Reddy. Survival analysis based framework for early prediction of student dropouts. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 903–912, 2016.
- [2] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370, 2012.
- [3] G. Balakrishnan and D. Coetzee. Predicting student retention in massive open online courses using hidden markov models. *Electrical Engineering and Computer Sciences University of California at Berkeley*, 53:57–58, 2013.
- [4] N. Blanchard, M. Brady, A. M. Olney, M. Glaus, X. Sun, M. Nystrand, B. Samei, S. Kelly, and S. D’Mello. A study of automatic speech recognition in noisy classroom environments for automated dialog analysis. In *International Conference on Artificial Intelligence in Education*, pages 23–33. Springer, 2015.
- [5] I. Borrelli, S. Caballero-Caballero, and E. Ponce-Cueto. Predict and intervene: Addressing the dropout problem in a mooc-based program. In *Proceedings of the Sixth (2019) ACM Conference on Learning@ Scale*, pages 1–9, 2019.
- [6] S. Boyer and K. Veeramachaneni. Transfer learning for predictive models in massive open online courses. In *International Conference on Artificial Intelligence in Education*, pages 54–63. Springer, 2015.
- [7] J. M. Bridgeland, J. J. DiIulio Jr, and K. B. Morison. The silent epidemic: Perspectives of high school dropouts. *Civic Enterprises*, 2006.
- [8] J. S. Catterall. On the social costs of dropping out of school. *The High School Journal*, 71(1):19–30, 1987.
- [9] J. Chen, H. Li, W. Wang, W. Ding, G. Y. Huang, and Z. Liu. A multimodal alerting system for online class quality assurance. In *International Conference on Artificial Intelligence in Education*, pages 381–385. Springer, 2019.
- [10] C. Coleman, R. S. Baker, and S. Stephenson. A better cold-start for early prediction of student at-risk status in new school districts. 2020.

- [11] C. A. Coleman, D. T. Seaton, and I. Chuang. Probabilistic use cases: Discovering behavioral patterns for predicting certification. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 141–148. ACM, 2015.
- [12] S. Crossley, L. Paquette, M. Dascalu, D. S. McNamara, and R. S. Baker. Combining click-stream data with nlp tools to better understand mooc completion. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 6–14. ACM, 2016.
- [13] V. Dupéré, E. Dion, T. Leventhal, I. Archambault, R. Crosnoe, and M. Janosz. High school dropout in proximal context: The triggering role of stressful life events. *Child development*, 89(2):e107–e122, 2018.
- [14] T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.
- [15] M. Fei and D.-Y. Yeung. Temporal models for predicting student dropout in massive open online courses. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 256–263. IEEE, 2015.
- [16] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.
- [17] J. Gardner, Y. Yang, R. S. Baker, and C. Brooks. Modeling and experimental design for mooc dropout prediction: A replication perspective.
- [18] N. Gitinabard, F. Khoshnevisan, C. F. Lynch, and E. Y. Wang. Your actions or your associates? predicting certification and dropout in moocs with behavioral and social features. *arXiv preprint arXiv:1809.00052*, 2018.
- [19] J. A. Greene, C. A. Oswald, and J. Pomerantz. Predictors of retention and achievement in a massive open online course. *American Educational Research Journal*, 52(5):925–955, 2015.
- [20] S. Halawa, D. Greene, and J. Mitchell. Dropout prediction in moocs using learner activity features. *Proceedings of the Second European MOOC Stakeholder Summit*, 37(1):58–65, 2014.
- [21] J. He, J. Bailey, B. I. Rubinstein, and R. Zhang. Identifying at-risk students in massive open online courses. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
- [22] T. K. Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282. IEEE, 1995.
- [23] S. Jiang, A. Williams, K. Schenke, M. Warschauer, and D. O’ Dowd. Predicting mooc performance with week 1 behavior. In *Educational Data Mining 2014*, 2014.
- [24] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.
- [25] J. Kim, P. J. Guo, D. T. Seaton, P. Mitros, K. Z. Gajos, and R. C. Miller. Understanding in-video dropouts and interaction peaks in online lecture videos. In *Proceedings of the First ACM Conference on Learning@ Scale Conference*, pages 31–40. ACM, 2014.
- [26] R. F. Kizilcec and S. Halawa. Attrition and achievement gaps in online learning. In *Proceedings of the second (2015) ACM Conference on Learning@ Scale*, pages 57–66. ACM, 2015.
- [27] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 170–179. ACM, 2013.
- [28] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein. *Logistic regression*. Springer, 2002.
- [29] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart. Predicting mooc dropout over weeks using machine learning methods. In *Proceedings of the EMNLP 2014 workshop on analysis of large scale social interaction in MOOCs*, pages 60–65, 2014.
- [30] J. E. Knowles. Of needles and haystacks: Building an accurate statewide dropout early warning system in wisconsin. *JEDM| Journal of Educational Data Mining*, 7(3):18–67, 2015.
- [31] D. Koller. Moocs on the move: How coursera is disrupting the traditional classroom. *Knowledge@ Wharton Podcast*, 2012.
- [32] J. E. Lansford, K. A. Dodge, G. S. Pettit, and J. E. Bates. A public health perspective on school dropout and adult outcomes: A prospective study of risk and protective factors from age 5 to 27 years. *Journal of Adolescent Health*, 58(6):652–658, 2016.
- [33] S. Lee and J. Y. Chung. The machine learning-based dropout early warning system for improving the performance of dropout prediction. *Applied Sciences*, 9(15):3093, 2019.
- [34] H. Li, Y. Kang, W. Ding, S. Yang, S. Yang, G. Y. Huang, and Z. Liu. Multimodal learning for classroom activity detection. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9234–9238. IEEE, 2020.
- [35] H. Li, Z. Wang, J. Tang, W. Ding, and Z. Liu. Siamese neural networks for class activity detection. In *International Conference on Artificial Intelligence in Education*. Springer, 2020.
- [36] Z. Liu, G. Xu, T. Liu, W. Fu, Y. Qi, W. Ding, Y. Song, C. Guo, C. Kong, S. Yang, and G. Y. Huang. Dolphin: A spoken language proficiency assessment system for elementary education. In *Proceedings of the Web Conference 2020*, page 2641–2647. ACM, 2020.
- [37] I. Lykourantzou, I. Giannoukos, V. Nikolopoulos, G. Mpardis, and V. Loumos. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers & Education*, 53(3):950–965, 2009.
- [38] J. McFarland, B. Hussar, C. De Brey, T. Snyder, X. Wang, S. Wilkinson-Flicker, S. Gebrekristos, J. Zhang, A. Rathbun, A. Barner, et al. The condition of education 2017. nces 2017-144. *National Center for Education Statistics*, 2017.
- [39] G. Mendez, T. D. Buskirk, S. Lohr, and S. Haag. Factors associated with persistence in science and engineering majors: An exploratory study using classification trees and random forests. *Journal of*

- Engineering Education*, 97(1):57–70, 2008.
- [40] M. Monrad. High school dropout: A quick stats fact sheet. *National High School Center*, 2007.
 - [41] A. M. Pallas. *School dropouts in the United States*. Center for Education Statistics, Office of Educational Research and . . . , 1986.
 - [42] F. D. Pereira, E. Oliveira, A. Cristea, D. Fernandes, L. Silva, G. Aguiar, A. Alamri, and M. Alshehri. Early dropout prediction for programming courses supported by online judges. In *International Conference on Artificial Intelligence in Education*, pages 67–72. Springer, 2019.
 - [43] A. Ramesh, D. Goldwasser, B. Huang, H. Daume III, and L. Getoor. Learning latent engagement patterns of students in online courses. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.
 - [44] C. P. Rosé, R. Carlson, D. Yang, M. Wen, L. Resnick, P. Goldman, and J. Sherer. Social factors that contribute to attrition in moocs. In *Proceedings of the First ACM Conference on Learning@ Scale Conference*, pages 197–198. ACM, 2014.
 - [45] R. W. Rumberger. High school dropouts: A review of issues and evidence. *Review of educational research*, 57(2):101–121, 1987.
 - [46] S. R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, 1991.
 - [47] M. Sharkey and R. Sanders. A process for predicting mooc attrition. In *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, pages 50–54, 2014.
 - [48] J. Son, I. Jung, K. Park, and B. Han. Tracking-by-segmentation with online gradient boosting decision tree. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3056–3064, 2015.
 - [49] R. Sullivan et al. Early warning signs. a solution-finding report. *Center on Innovations in Learning, Temple University*, 2017.
 - [50] C. Tang, Y. Ouyang, W. Rong, J. Zhang, and Z. Xiong. Time series model for predicting dropout in massive open online courses. In *International Conference on Artificial Intelligence in Education*, pages 353–357. Springer, 2018.
 - [51] C. Taylor, K. Veeramachaneni, and U.-M. O’Reilly. Likely to stop? predicting stopout in massive open online courses. *arXiv preprint arXiv:1408.3382*, 2014.
 - [52] Y. Wang. Exploring possible reasons behind low student retention rates of massive online open courses: A comparative case study from a social cognitive perspective. In *AIED 2013 Workshops Proceedings Volume*, page 58. Citeseer, 2013.
 - [53] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen. Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, 16(4):449–475, 2013.
 - [54] J. Whitehill, K. Mohan, D. Seaton, Y. Rosen, and D. Tingley. Mooc dropout prediction: How to measure accuracy? In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 161–164. ACM, 2017.
 - [55] L. Wood, S. Kiperman, R. C. Esch, A. J. Leroux, and S. D. Truscott. Predicting dropout using student-and school-level factors: An ecological perspective. *School Psychology Quarterly*, 32(1):35, 2017.
 - [56] W. Xing, X. Chen, J. Stein, and M. Marcinkowski. Temporal predication of dropouts in moocs: Reaching the low hanging fruit through stacking generalization. *Computers in Human Behavior*, 58:119–129, 2016.
 - [57] S. Xu, W. Ding, and Z. Liu. Automatic dialogic instruction detection for k-12 online one-on-one classes. In *International Conference on Artificial Intelligence in Education*. Springer, 2020.
 - [58] D. Yang, T. Sinha, D. Adamson, and C. P. Rosé. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Proceedings of the 2013 NIPS Data-driven education workshop*, volume 11, page 14, 2013.
 - [59] L. Zhang and H. Rangwala. Early identification of at-risk students using iterative logistic regression. In *International Conference on Artificial Intelligence in Education*, pages 613–626. Springer, 2018.
 - [60] S. Zhang, M. Lei, Z. Yan, and L. Dai. Deep-fsmn for large vocabulary continuous speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5869–5873. IEEE, 2018.

Erroneous Answers Categorization for Sketching Questions in Spatial Visualization Training

Tiffany Wenting Li
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
wenting7@illinois.edu

Luc Paquette
Department of Curriculum & Instruction
University of Illinois at Urbana-Champaign
Champaign, Illinois 61820
lpaq@illinois.edu

ABSTRACT

Spatial visualization skills are essential and fundamental to studying STEM subjects, and sketching is an effective way to practice those skills. One significant challenge of supporting practice using sketching questions is the vast number of possible mistakes, making it time-consuming for instructors to provide customized and actionable feedback to students. The same challenge persists for computer programs as well. This paper introduces a clustering model designed to categorize sketching answers based on the severity and characteristics of their mistakes. The model is designed to be used by a computer-based training platform to provide customized, actionable formative feedback to students in real-time. The promising results also suggest a new and comprehensive set of evaluation criteria to assess a student's performance on sketching questions. As a broader contribution, our work is a proof-of-concept for a modeling approach to automatically evaluate and provide formative feedback on complex free-hand sketches using abstract features that may be generalized to a variety of disciplines that involve the creation of technical drawings.

Keywords

Automatic grading, Sketching, Clustering, Spatial Visualization, Formative feedback

1. INTRODUCTION

Spatial visualization is the ability to represent and mentally manipulate two-dimensional and three-dimensional objects [11]. A body of research has shown that good spatial visualization skills help students succeed in STEM education [39, 3, 13, 25, 27, 32, 41, 44]. It is encouraging that existing research also demonstrates that spatial visualization skills are malleable and can be trained and improved, for example, via forms of workshops and seminars [42]. There have been successes in increasing the retention rates of STEM freshmen students with spatial visualization skills training in recent years, especially for minority groups such as female

students [39, 23].

Besides multiple-choice questions that are traditionally used in spatial visualization training, free-hand sketching on grid paper is an effective type of practice question [38]. Sketching questions can imitate the sketching tasks required in many engineering disciplines, which is particularly helpful since sketching is a fundamental skill for engineering designs [22]. In the training process, since students gain from learning from their mistakes instead of failing in the first try and giving up based on the immediate-feedback assessment technique [26], students can benefit from having a second chance on a practice problem. However, providing formative feedback while not giving away the answer, which is known to support self-regulated learning [28], on free-hand sketching can be challenging due to the wide variety of possible incorrect answers on such activities.

While human instructors possess the capability to analyze an erroneous free-hand sketch, identify the source of potential errors and provide formative feedback, it is a time-consuming process and providing such feedback to a large student population would require prohibitive efforts that would likely prevent the feedback from being provided in a timely fashion [2]. Computer-based systems able to provide timely formative feedback can be considered as an alternative to address this limitation. However, one significant challenge to automatically providing immediate customized feedback for sketching questions is the need for a computer-based system to be able to recognize and understand how much an answer is different from the answer key and the types of mistakes students are making.

On the one hand, sketching questions have an enormous number of possible incorrect answers, which are often specific to a unique problem, making it difficult, if not impossible, to identify every possible error and to prepare unique feedback for each one. As an alternative, a computer-based system could be designed to recognize categories of answers based on the severity or characteristics of their errors and provide feedback relevant to each one. However, to the best of our knowledge, there is no existing research that categorizes answers to complex sketching questions based on their errors, either conceptually or computationally. The lack of solution motivated us to identify patterns that exist in students' erroneous sketching answers and create a computer-based algorithm that can categorize them in real-time.

Tiffany Wenting Li and Luc Paquette "Erroneous Answers Categorization for Sketching Questions in Spatial Visualization Training" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 148 - 158

Due to the lack of existing categories of erroneous answers in free-hand sketching problems, we propose the use of a clustering approach to identifying such categories. Our research questions are the following:

- RQ1 What categories exist in students' sketching answers based on the severity and characteristics of their errors?
- RQ2 How meaningful are the identified erroneous answer categories, and what actionable feedback can be provided for each category?

We constructed a list of features that can be used to characterize students' erroneous sketching answers. Using a k-mean clustering approach, we discovered six common answer categories for incorrect sketches that are distinct from one another according to the severity and characteristics of the errors. Our clustering results suggest a new set of evaluation criteria for complex free-hand sketching answers that is more interpretable and generalizable than those in prior work [7, 43, 5]. Also, we provide initial suggestions for the kinds of formative feedback appropriate for each answer category without giving away the answer [36].

To the best of our knowledge, our study is the first to identify categories of erroneous sketches, both computationally and conceptually, in spatial visualization sketching problems using abstract features. Our approach also has the potential to be generalized to other subject areas that require sketching practices, mostly technical drawings in various Engineering and Science subjects, such as circuit diagrams in Electrical Engineering, engine models in Mechanical Engineering, building plans in Architecture, and structural formula in Organic Chemistry.

2. RELATED WORK

2.1 Spatial Visualization Skills and Sketching

Spatial visualization skills were estimated to play an important role in 84 careers [37], most of which are STEM-related. A longitudinal study showed that psychometrically-assessed spatial ability predicts career in STEM fields after accounting for Math and Verbal aptitudes [45].

Spatial visualization skills are applied in various STEM areas. Research shows that students with better spatial visualization skills perform better in Chemistry [32, 6]. In Organic Chemistry, for example, students with strong spatial visualization skills draw preliminary figures more often. Hence they use figures to gain a better understanding of the questions and are more likely to answer them correctly [32]. Another body of research revealed the connection between spatial skills and Geoscience [17, 30]. In particular, students with strong visual penetration ability, e.g., imagining cross-sections, perform better in Geology [17]. Furthermore, understanding cross-sectioning is a basic skill in many other engineering subjects [9, 12]. Spatial visualization is also found to be tightly related to performance in Anatomy in Biology [34], Radiology in Medicine [16].

A wide variety of empirical research has shown that spatial visualization skills are malleable. Interventions designed

to improve spatial visualization skills reach, on average, a medium effect size of 0.47 [42]. A well-known training developed by Sorby (2009) showed significant post-test improvement for each class of college students over a 6-years-long study. In particular, Sorby found that the training significantly improved female students' retention rate but not that of male students [39]. The finding suggested the critical role of spatial visualization skills training in increasing the diversity of STEM field students.

Sketching ability is fundamental to engineering design [22] and highly correlates with many STEM subjects [35]. To improve spatial visualization skills, sketching is one of the most effective approaches [38]. Electronic sketching has also demonstrated potential in training spatial visualization skills [8, 47]. Thus, the application of sketching practice is worth studying for better improving spatial visualization skills.

2.2 Computer-based Evaluation and Formative Feedback for Sketches

To the best of our knowledge, there is no prior work on the evaluation of sketches in spatial visualization training, both conceptually or computationally. The use of computer-based formative feedback for spatial visualization sketching has not been studied either. There is a body of research on computer-based evaluation and formative feedback for other types of sketches [5, 7, 43, 40, 15, 18, 19, 20]. However, some of them are too simple or too domain-specific to be generalized to a complicated case as in spatial visualization sketches. Others' evaluation methods cannot provide actionable or easy-to-interpret formative feedback.

For free-hand sketching that is evaluated mostly based on the shape and structure, there are a few existing evaluation approaches in domains other than spatial visualization training. Bhat (2017) developed Skechography, a river-sketching auto-grading tool for Geology [5]. This tool could perform sketch recognition and compare the river's shape similarity using the Shape Context algorithm, the distances of start points and endpoints between a student's answer and the answer key. Based on the degree of similarity and distances, the tool provided a score that was a weighted sum of these three features. Skechography evaluated a river, which had only one line with specific features of a start point, an endpoint, and the shape of the line. The simplicity of this application has a weak external validity and cannot be used in evaluating spatial visualization sketches.

The work by Chandan et al. (2018) [7], on the other hand, worked on a complicated case of free-hand drawing of objects of specific categories, e.g., a bee, an airplane, etc. They applied a Convolutional Neural Network approach for object categorization and a Scale Invariant Feature Transform approach to check the similarity between a given sketch and the "standard" sketch. As feedback, the tool showed the percentage of similarity to various categories of objects. The use of deep learning methods made the interpretation of results challenging. Hence, this approach is limited in its capability to generate specific and actionable feedback to help students improve their answers.

Mechanix, a sketch-based tutoring system for learning forces applied on a truss, could provide specific feedback to free-

hand sketching of forces [43]. In this case, the errors that could occur were known and clearly defined on an arrow-basis. Given the small number of arrows, it is relatively easy to cater specific and actionable feedback to each error. In the case of spatial visualization sketches, a sketch contains far more number of lines, making it infeasible to provide a piece of feedback for each line.

There exists another body of work that focused on the recognition of East Asian characters, which are similar to a simple sketch [40]. However, these solutions applied an "all or nothing" approach to recognize the structure of a character, which was not helpful in providing specific formative feedback. A few other works aimed to evaluate and provide feedback on the quality or aesthetics of a sketch, but not on the correctness in terms of the structure of shape [15, 18]. There is also an evaluation approach for computer-aided design solid models specifically, using criteria related to parameters set in the computer-aided model, which does not apply to free-hand sketching because the concept of parameters is not intuitive in free-hand sketching [19, 20].

Overall, there is limited work on a computer-based evaluation of complex free-hand sketching based on structural correctness that can generate specific and actionable formative feedback. Our work aims to fill in this gap.

2.3 Answers Categorization in Content-based Automated Evaluation

In evaluating constructed response automatically from a content-based perspective, there is a rich body of work in evaluating short answer questions for a variety of subjects and domains [24]. However, except for the studies mentioned in the last section, there is very few existing literature related to the content-based evaluation of complex free-hand sketching. Therefore, we draw our inspiration from the existing research in evaluating short answer questions and apply it to complex free-hand sketches, a different type of constructed response.

Answer categorization is one of the most frequently used approaches to perform a content-based evaluation of short answers. In most cases, supervised learning is applied using a manually labeled training set based on pre-defined rubrics [21, 33, 1, 10, 29]. For example, c-rater applied NLP techniques that determined whether an answer contained each key concept and was widely applied on short answer questions in Biology, Psychology, Math, and Reading, to not only grade but to provide specific real-time feedback [21, 1]. Pulman and Sukkari (2005) experimented with Inductive Logic Programming, Decision Tree and Naive Bayes to classify short answers into the desired category for Biology [33].

In our case, however, there are neither pre-existing robust rubrics as the evaluation standard for spatial visualization sketches nor known categories of error. This brought difficulties to label a training set manually accurately. Also, most content-based evaluation approaches only provided up to three levels of scoring. Some exceptions that provided more than three levels of scoring were either unclear about the definition of the levels or the levels were only mechanical composition of the correct answer [24]. As an alternative,

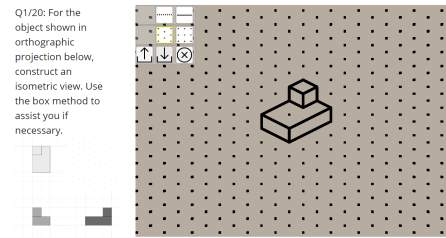


Figure 1: Free-hand sketching tool for isometric sketching on the online spatial visualization training platform

we turned to unsupervised learning to perform answer categorization to identify categories that were as granular yet meaningful as possible. Clustering is an often-used unsupervised learning approach in short-answer grading, especially in the case of answering open-ended questions. Previous work [4, 48] has shown that clustering could group answers that are similar in text characteristics, semantics, and topics. Our work aims to leverage this method to categorize complex sketches in spatial visualization training.

3. METHODS

3.1 Data Collection

We collected data from students solving free-hand sketching problems in a 100-level engineering course called "Spatial Visualization" that utilized an online training platform over half a semester in Fall 2019 at our home institution, a large public university in the Midwestern United States. The online training platform was previously developed as a computer-based spatial visualization training platform [47] to enable practicing at scale using online exercise and automatic grading. Previous work has shown a significant improvement in spatial visualization skills for those who completed the exercises on the platform [47].

Students in the course met once a week in-person for an hour, and the majority part of the course was working through practice problems on the platform on their own as their weekly assignment, given the instructions. The focus of practice questions each week was different, depending on the particular set of skills that were being trained, such as mental rotation, cross-sectioning, and coded plan. The platform supports both multiple-choice questions and sketching questions. Figure 1 and Figure 2 show the free-hand sketching tool on the platform that allows students to sketch out their answers on the computer. Students can draw and erase lines on the grid paper freely. Students could also save their sketch when they leave the platform and load what they saved when they come back. In the course, students were given a maximum of two attempts for each sketching question, i.e., they were given a second chance if they answered incorrectly in the first attempt. All the sketching questions were graded with an "all or nothing" approach.

The collected dataset includes 370 incorrect sketches from 14 students in the course that covers five types of sketching questions and 61 unique questions. We excluded correct sketches in the categorization because they would naturally

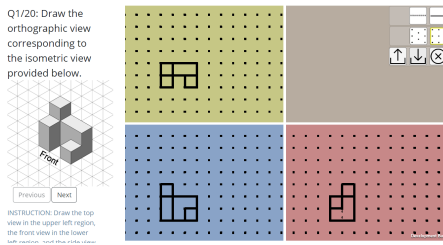


Figure 2: Free-hand sketching tool for orthographic sketching on the online spatial visualization training platform

be in one category by mapping exactly to the answer key. Examples of the types of sketching questions include drawing the orthographic view of a 3D object given the isometric view or vice versa, and drawing the resulting 3D object after rotating a given 3D object with a certain degree in a given direction. Each type of sketching questions contained a series of different questions with 3D objects of various shapes. On average, each sketch contains approximately 30 to 80 lines of unit length.

Each submission of an attempt to answer a question produced a raw log describing their answer. In the raw log, two major types of information were recorded. First, it contained the set of lines in the final submitted sketch. Second, it recorded the history of all the timestamped steps a student took of adding or deleting a line, clearing, or loading the sketch for that question (Figure. 3). In this paper, we focused on the final submitted sketch only since the goal is to categorize the final answer instead of analyzing students' process of solving a free-hand sketching problem.

Each final submitted sketch is represented by the X-Y coordinates of a list of lines. The lines are further denoted by the type of the lines, either solid line or dashed line, which are the two standard types of lines used in the sketching exercise for different purposes. A sketch is mostly made up of solid lines, but a dashed line should be used instead of a solid line to represent a hidden edge from a particular perspective.

Another data point in the raw log is the type of grid paper used for a sketch. There are two types of grid paper in the sketching exercises: an isometric grid for isometric drawing, and a dot grid for orthographic drawing. A sketch is considered as correct only if the shape and the size of the object match with those of the answer key, and uses the correct type of grid paper. The position of where a sketch is drawn on the grid paper is flexible.

We performed two steps of data standardization on the raw log before feature extraction. First, we aligned both the student's answer and the answer key to the lower-left corner of the sketch-pad. Second, all the lines were broken down into unit length and de-duplicated so that lines that overlapped with each other would only be counted once. We conducted these two steps for the ease of comparing student's answers against the answer key.

3.2 Feature Extraction

```
GridISO 18
SolidLine: "13, 11, 15, 10" "15, 10, 18, 12" "18, 12, 17, 12" "17, 12, 16, 12" "16, 12, 15, 12" "15, 12, 13, 11" "13, 11, 13, 18" "13, 18, 15, 9" "15, 9, 18, 11" "18, 11, 18, 12" "15, 10, 15, 9" "15, 12, 15, 13" "15, 13, 16, 13" "16, 12, 16, 13" "16, 13, 17, 13" "17, 13, 17, 12" "15, 13, 16, 14" "16, 14, 17, 13"
DashedLine:
##### History List #####
#####
##HistoryItem## Type: ChangeGridTypeToISO, Line: (0,0,0,0), Time: 10/26/2019 10:33:06
##HistoryItem## Type: LineAdd, SolidLine: (13,11:15,10), Time: 10/26/2019 10:33:10
##HistoryItem## Type: LineAdd, SolidLine: (15,10:18,12), Time: 10/26/2019 10:33:12
##HistoryItem## Type: LineAdd, SolidLine: (18,12:17,12), Time: 10/26/2019 10:33:13
##HistoryItem## Type: LineAdd, SolidLine: (17,12:16,12), Time: 10/26/2019 10:33:14
##HistoryItem## Type: LineAdd, SolidLine: (16,12:15,12), Time: 10/26/2019 10:33:15
##HistoryItem## Type: LineAdd, SolidLine: (15,12:13,11), Time: 10/26/2019 10:33:15
##HistoryItem## Type: LineAdd, SolidLine: (13,11:13,10), Time: 10/26/2019 10:33:16
##HistoryItem## Type: LineAdd, SolidLine: (13,10:15,9), Time: 10/26/2019 10:33:17
##HistoryItem## Type: LineAdd, SolidLine: (15,9:18,11), Time: 10/26/2019 10:33:18
##HistoryItem## Type: LineAdd, SolidLine: (18,11:18,12), Time: 10/26/2019 10:33:18
##HistoryItem## Type: LineAdd, SolidLine: (15,10:15,9), Time: 10/26/2019 10:33:19
##HistoryItem## Type: LineAdd, SolidLine: (15,12:15,13), Time: 10/26/2019 10:33:21
##HistoryItem## Type: LineAdd, SolidLine: (15,13:16,13), Time: 10/26/2019 10:33:22
##HistoryItem## Type: LineAdd, SolidLine: (16,12:16,13), Time: 10/26/2019 10:33:24
##HistoryItem## Type: LineAdd, SolidLine: (16,13:17,13), Time: 10/26/2019 10:33:25
##HistoryItem## Type: LineAdd, SolidLine: (17,13:17,12), Time: 10/26/2019 10:33:26
##HistoryItem## Type: LineAdd, SolidLine: (15,13:16,14), Time: 10/26/2019 10:33:28
##HistoryItem## Type: LineAdd, SolidLine: (16,14:17,13), Time: 10/26/2019 10:33:29
##HistoryItem## Type: ProjSave, Line: (0,0,0,0), Time: 10/26/2019 10:33:32
##HistoryItem## Type: ProjSave, Line: (0,0,0,0), Time: 10/26/2019 10:34:12
```

Figure 3: An example of a raw log file generated from sketching questions on the online spatial visualization training platform

We developed a total of 8 features to use as input for our clustering model. We performed feature engineering manually after observing a small subset of the data to get an idea of what information human instructors might use when interpreting incorrect answers. In order to get a preliminary view of possible errors that would be as comprehensive as possible, we selected three questions that had the highest number of incorrect answers and observed the errors made by students on those problems. Based on our preliminary observation, we created three categories of features that represent different characteristics of the observed errors.

The first group of features uses a unit-length line as its basic unit, i.e., a line connecting adjacent points, and represents the number of lines that are wrong compared to the answer key. We observed from the subset of mistakes that the number of incorrect lines involved in a sketch varied widely, from only one wrong line to over 80% of lines being wrong. The number of incorrect lines is a straightforward way to quantify the degree to which a sketch was incorrect. We considered three scenarios in which a line is wrong.

1. **An extra line:** a line is in the student's answer, but there is no line at the same position in the answer key.
2. **A missing line:** a line is in the answer key, but there is no line at the same position in the student's answer.
3. **A line with incorrect type:** two lines with the same position in the student's answer and the answer key are of different types, i.e., solid line vs. dashed line.

To normalize the number of incorrect lines against the complexity of the sketch, we adopted the percentage of wrong lines instead of the absolute number, i.e., dividing by the total number of lines in a sketch. The three features in this group are Percentage of Extra Lines, Percentage of Missing Lines, and Percentage of Lines with Correct Position but Incorrect Type.

The second category of features represents the groupings of the incorrect lines based on their location in a sketch. In our preliminary observation, we found that, between two

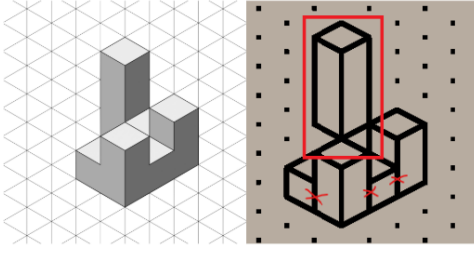


Figure 4: An example of a sketch (on the right) with four error components, i.e., four sites of mistakes. The sketch on the left is the answer key.

sketches with a similar number of incorrect lines, the incorrect lines may be inter-connected and concentrated in one place in a sketch while being scattered in multiple spots in another sketch. These two cases represented the mistakes of different natures.

Based on the assumption that incorrect lines that are connected are more likely caused by the same mistake, we treated all the incorrect lines as an undirected graph and defined each component in the graph as one "site" of mistake. A component here has the same definition of a component in an undirected graph, a subgraph in which any two vertices are connected by paths, and which is connected to no additional vertices in the supergraph [46]. As an example, in Figure 4, there are a total of four error components in the sketch, three extra lines in different locations, and a disconnected taller stack separated from the bottom of the object.

We constructed three features in this category. The first feature is the number of components in the graph made of incorrect lines, which is a representation of the number of mistake sites in a sketch. Since the size of a component represents how severe a mistake is, the second feature is the average size of all the error components in a sketch. The larger the average component size is, the more severe the mistakes are on average. The last feature is the maximum size difference among all error components, which reflects the range of severity across multiple mistake sites in a sketch.

The last set of features describes the general characteristics of the sketch. One feature is whether the student uses the same type of sketching grid as the answer key. Another feature is whether the sketch is empty. If it is empty, it indicates either the student did not attempt the question or accidentally skipped the question.

3.3 Model Construction

As there was no prior framework or knowledge on how to categorize the erroneous sketches, it was not possible to obtain labels (ground truth) describing each answer. As such, we used an unsupervised clustering algorithm to identify categories of erroneous answers from existing data. Based on prior observation of the data, we hypothesized that the features of each cluster should have a sphere-like shape. Therefore, we used k-means clustering with squared Euclidean distance. The algorithm aims to assign all the data points into a specified number of clusters such that every data point is

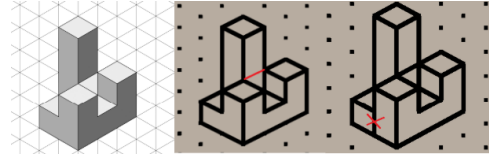


Figure 5: Examples of mistakes in Cluster 0, having one minor mistake. The sketch on the left is the answer key.

in the cluster with the nearest mean. Ideally, data points that have similar values across all the features are grouped in one cluster.

After feature extraction, we performed further data normalization as the first step of model construction. Since the k-means clustering algorithm is sensitive to the scale of the features, we normalized each of the three features (Number of Components, Average Size of Components, and Maximum Difference between Size of All Components) into the unit interval respectively across all data, so that they were on the same scale as the other features that were either in percentages or in a boolean format.

We performed parameter tuning to decide on the optimal number of cluster k . We started with two clusters and repeatedly increased the number of clusters by one. We evaluated the choice of k using two criteria. The main criterion we used to evaluate the quality of the clustering results was how interpretable a new cluster was and whether it could help us provide more specific and actionable feedback. Another complementary criterion for evaluation was the Silhouette score, measuring the quality of the clusters based on the cohesion of the separation of the identified clusters (Silhouette score ranges from -1 to 1). We valued the interpretability of a cluster over a higher Silhouette score. Therefore, as long as the Silhouette score remained at an acceptable level, we increased k until the interpretation of the newly generated cluster did not make sense or did not differ much from the existing clusters.

4. RESULTS

Our clustering approach identified a set of six clusters related to categories of erroneous answers in free-hand sketching problems, as listed in Table 1. The 6 clusters are ordered based on the severity of the errors in the table. The clustering model yields a Silhouette score of 0.6659, which is a reasonable value.

Cluster 0 is the most common cluster in the dataset. From the centroid value, we can see that the sketches in this cluster only have one mistake (Number of Component = 1) with about two incorrect lines (Avg Component Size = 1.89). The centroid values suggested that a large portion of the errors had only one minor mistake, which was most likely due to drawing errors such as forgetting an edge at the corner, or drawing an extra edge on a plane (see examples in Fig 5).

Cluster 1, the second-largest cluster in the dataset, differs from Cluster 0 mainly by the number of mistakes in the sketch. On average, there are 2.21 mistake components in

Cluster ID	Cluster Size	Interpretation	Perc Missing	Perc Extra	Perc Type	Num Comp	Avg Comp Size	Max Size Diff	Same Grid?	Empty?
0	218	Have one minor mistake	2.39%	2.16%	0.04%	1.00	1.89	0.00	1.00	0.00
1	65	Have more than one minor mistakes	4.13%	10.46%	0.11%	2.14	2.88	1.43	1.00	0.00
2	30	Have both major and minor mistakes, mostly minor mistakes	20.61%	32.14%	0.29%	3.70	5.13	5.63	1.00	0.00
3	15	Have both major and minor mistakes, mostly major mistakes	37.82%	22.46%	0.77%	2.80	10.69	15.73	1.00	0.00
4	39	More than half of the sketch as a whole is completely wrong	80.08%	67.04%	0.00%	1.05	45.35	0.08	1.00	0.00
5	3	Empty sketch	100.00%	0.00%	0.00%	1.67	29.78	1.00	0.33	1.00

Table 1: Clustering Results Summary Table: The size, interpretation and centroid of each cluster are shown in the table. The centroid values are transformed back to its original scale if unit normalization was performed. Values are color-coded with different shades of red, representing low values to high values)

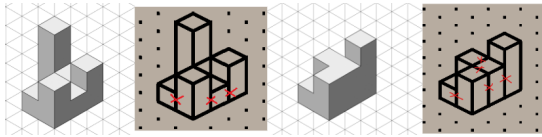


Figure 6: Examples of mistakes in Cluster 1, having multiple minor mistakes. The sketches with a white background are the answer keys.

the sketch. The average size of 3.04 lines of the components suggests that these are still minor mistakes with three incorrect lines on average. It is reasonable to interpret Cluster 1 as sketches that have several minor mistakes. Examples of this category are shown in the examples in Fig 6. Even though both Cluster 0 and Cluster 1 contain minor errors, they are different enough because students in Cluster 0 make one small mistake likely due to being careless. In contrast, those in Cluster 1 may have misconceptions that are causing a series of mistakes.

Cluster 2 and 3 are quite different from Cluster 0 and Cluster 1. Both of them have a much higher Percentage of Missing Lines and Percentage of Extra lines compared to Cluster 0 and 1, suggesting more severe mistakes in the sketch. More severe errors are more likely to be due to an incorrect structure at specific parts of the sketch rather than careless mistakes. These two clusters both have a high number of components (3.70 and 2.80 for Cluster 2 and 3 respectively), suggesting a series of mistakes across the sketch. Cluster 2 and 3 are different in two perspectives. First, Cluster 2's average component size is small (5.13), while Cluster 3's average component size is a lot bigger (10.69). Second, Cluster 3 has a massive difference in size across the different components (15.73), while Cluster 2 has a medium difference of 5.63. These differences suggest that within the series of mistakes in a sketch in Cluster 2, more of them are minor,

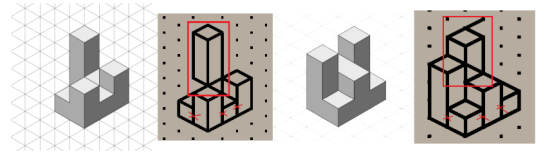


Figure 7: Examples of mistakes in Cluster 2, having multiple minor mistakes and a small number of major mistakes. The sketches with a white background are the answer keys.

and there is only a small proportion of major mistakes, as shown in Figure 7. On the other hand, a sketch in Cluster 3 has mainly major mistakes and fewer minor mistakes, as shown in Figure 8. The major mistakes in Cluster 3 are also more severe than those in Cluster 2 on average.

Cluster 4 has 80% of the lines missing and 67% extra lines, a lot higher than the previous clusters. Interestingly, most of the sketches in this cluster have only one component in their mistake (1.05 components on average), with an average size of 45.35 lines. These features suggest that there is one substantial mistake that spans over half of the sketch, which is often due to either an utterly wrong structure or a wrong orientation. For example, both examples in Fig 9 have the correct structure but wrong orientations.

Lastly, Cluster 5 contains empty answers, either due to the student not attempting a question or accidentally skipping it. Even though the cluster size is small, with only 3 data points due to the low number of empty answers, it is distinct enough from all the other clusters to be on its own.

Overall, we considered the erroneous answer categories detected to be intuitive and well-defined. They are distinct in the severity and characteristics of the mistakes. Being able

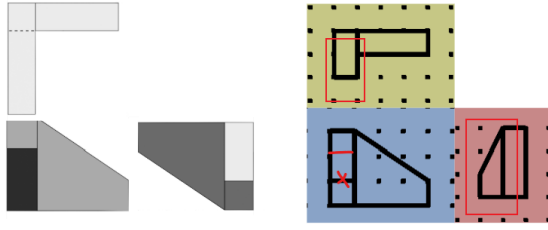


Figure 8: An example of mistake in Cluster 3, having multiple major and minor mistakes, but mainly major mistakes. The sketch on the left is the answer key.

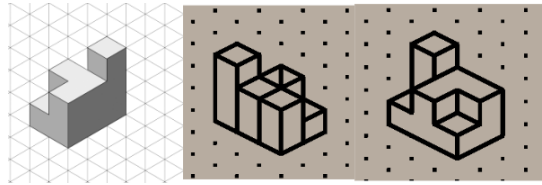


Figure 9: Examples of mistakes in Cluster 4, having one huge cluster of mistake. The sketch with a white background is the answer key.

to automatically identify six categories of erroneous answers demonstrated the potential advantage of using an unsupervised approach in answer categorization than a supervised learning approach that tries to align the model capability with human judgment of the answer categories, which could often only yield up to three clearly defined categories [24]. Additionally, we did not observe any significant difference between the frequency distribution of the error categories across the different types of questions in our dataset, i.e., the frequency of each answer category did not differ significantly across all five types of sketching questions, suggesting the generalizability of the error categories to more variety of questions.

5. DISCUSSION

5.1 Evaluation Criteria for Sketching

Due to the lack of prior work on erroneous answer categories in complex free-hand sketching problems, there is no currently available set of criteria to evaluate the degree of correctness of a complicated sketching answer. In multiple offerings of the spatial visualization training in the past in our school, an instructor either used an "all or nothing" evaluation approach, or used a subjective standard on one or two dimensions to judge a sketch, e.g., taking off 0.5 point for each missing or extra line up to a maximum of 1 point, taking off 1.5 points any time when not all features of the top, front, and right sides are correct. These evaluation schemes are too coarse to reflect the degree of correctness of a sketch accurately. The results of our clustering analysis provide promising results towards the development of a more comprehensive view on how to evaluate a sketch using a scale of multiple levels.

Our model demonstrated that more than one dimension is needed concurrently to provide a nuanced interpretation of the state of a sketch. In our model, the percentages of missing, extra lines or lines with the wrong type, the number of mistakes sites, the average size of the mistakes, and how different the various mistakes sites are in a sketch are used in combination with one another to determine the degree of correctness and the type of errors. For example, a distinction between Cluster 2 and 3 suggests that with a similar percentage of incorrect lines, the number of mistakes components and the average size of the components brings additional insights into whether a sketch contains a large number of minor mistakes or a small number of major mistakes. As another example, even though Cluster 0 and Cluster 1 have a similar average size of mistakes, the number of mistake sites suggests that students in Cluster 1 may have a more systematic misconception than those in Cluster 0 who likely commit a mistake due to carelessness.

Our approach could also be used to define minor mistakes versus major mistakes in a sketch for a group of sketching questions with similar size and complexity. Without a systematic review of all the mistakes in a group of sketching questions, it is hard for an instructor to draw an objective line between an error that is significant and one that is not. As a result, the evaluation criteria may be overly strict or overly generous. The clustering model computationally categorizes what it considers as minor and major mistakes based on the optimal separation principle. Its outcome can serve as analytical support for an instructor's grading decision.

5.2 Potential Intervention

Since one of the motivations to construct this model is to provide real-time, customized, and actionable formative feedback, we propose potential customized intervention messages for each erroneous answer category. Based on the best practices of offering formative feedback [36], each of the messages follow a similar structure of (1) first letting the student know how far they are from the correct answer, (2) describing what types of mistake there are, and (3) suggesting ways for the student to approach solving the errors. A summary of the interventions is provided in Table 2.

Students having answers that fall into Cluster 0 or Cluster 1, which consist of having one or more minor errors, understand what the object should look like structure-wise. When the system tells them that they are wrong, they may find it confusing since they are likely confident in their answer. Hence, the feedback message could first assure the students that they have got the general structure of the object correct. Then, the system could let the students know that they have X number of minor mistakes, where X is the feature Number of Components. The feedback may also include whether they have some missing lines, extra lines, or lines of the wrong type. Lastly, the feedback message would suggest the students check for details of their drawing by listing out the common reasons for such errors, such as extra edges on a flat plane, missing edges at a corner.

If the answer falls within Cluster 2 or Cluster 3, the feedback message should be different from that for Cluster 0 and 1 because there is at least one major mistake in the answer,

Cluster ID	Cluster Size	Interpretation	Potential Intervention
0	218	Have one minor mistake	<ul style="list-style-type: none"> Encourage students that they get the general structure correct Inform students the number of minor mistake sites they have Suggest students to check for detail errors and list the common reasons for such errors, e.g. extra edges on a flat plane, missing edges at a corner
1	65	Have more than one minor mistakes	
2	30	Have some major and minor mistakes, mostly <u>minor</u> mistakes	<ul style="list-style-type: none"> Encourage students that they are heading towards the right direction Inform students the number of minor and major mistake sites they have Suggest students to revisit some parts of the structure Suggest students to carefully check for drawing errors and list the common reasons for such errors, e.g. extra edges on a flat plane, missing edges at a corner
3	15	Have some major and minor mistakes, mostly <u>major</u> mistakes	
4	39	More than half of the sketch as a whole is completely wrong	<ul style="list-style-type: none"> If students have the correct structure but a wrong orientation: <ul style="list-style-type: none"> Encourage students that they get the general structure correct Inform them that they may have drawn it in an incorrect orientation If students have an incorrect structure: <ul style="list-style-type: none"> Let students know that they have the wrong idea for the structure Suggest students to rethink about the structure from the beginning Provide hints for the students if available
5	3	Empty sketch	<ul style="list-style-type: none"> If students did not make an effort, encourage them to attempt the question If students forgot to submit a sketch, remind them to submit in the next attempt

Table 2: Interventions Summary Table

likely due to a structural error. The students in these two cases are mostly on the right track in terms of the general structure of the sketch. Hence, the feedback message could first encourage them that they are heading in the right direction. The system could then say that the sketch has X minor mistakes and Y major mistakes, where X is the Number of Components with a size smaller than the Average Component Size of the cluster centroid, and Y is the Number of Components with a size larger than the average. Finally, the intervention message could suggest the student first revisit the structure in detail to identify the major mistake, and then carefully check for drawing errors referring to a list of common minor mistakes.

For a student that falls into Cluster 4, it is likely that the student is either on the wrong track entirely or uses a wrong orientation. The system can perform a further check to compare the student's answer to other possible orientations and see if it belongs to the case of having a wrong orientation. If it is, the feedback message will remind the student that the structure of the sketch is mostly correct, but the orientation is incorrect. If it is not the case of having a wrong orientation, the feedback message will remind the students that they may have the wrong idea for the sketch, and they should reconsider the question from the beginning. The system could consider providing hints to the students as well in this case.

Lastly, if a student submits an empty sketch, the system can check the time spent on the question to determine whether the student did not attempt the question at all or forgot to click the submit button. If the student did not attempt the question, the system would encourage the student to make an effort in attempting to solve the problem. If the student forgot to submit the answer, the feedback message would

remind them to submit in the next attempt.

5.3 Generalizability of the Proof-of-concept Approach

Our clustering model is more than a single model that works only in a specific scenario. It is a proof-of-concept approach for the evaluation of a complex free-hand sketch based on abstract features. Our contributions to the evaluation scheme of sketching answers have the potential to be generalized from spatial visualization training to more fields that involve free-hand technical drawings in various Engineering and Science subjects, such as circuit diagrams in Electrical Engineering, engine models in Mechanical Engineering, building plans in Architecture, and structural formula in Organic Chemistry. Technical drawing is similar to spatial visualization sketching in the sense that they both follow strict rules of sketching and are often drawn on grid paper to ensure a consistent proportion and orientation. Technical drawings in these fields usually start from a fundamental practice of drawing and modeling using practice problems that have a limited number of correct answers. With the presence of answer keys, our unsupervised clustering approach is flexible and easy to be retrained on new datasets to adapt to new types of sketches, even with additional features developed based on the learning goal of the type of sketches.

On the other hand, for technical drawing that involves a creative component or pure creative drawing, it may be harder to apply our approach directly. In evaluating creative drawing that does not have a limited number of correct answers, a mistake may be more subjective, and the evaluation may extend beyond getting a sketch correct to being functional, optimal, creative or aesthetic. The clustering approach based on abstract features of a sketch, however, may be used for

other purposes in this case. For example, our approach could be used to group sketches with similar characteristics together for the convenience of human graders, especially in a large course with limited human resources, such as Massive Open Online Courses. Reconsideration in feature engineering would be needed to achieve the new goals.

6. LIMITATIONS AND FUTURE WORK

The current erroneous answer categories do not take into account specific reasons that lead to a particular error in an answer. There may be multiple reasons for a student to end up with mistakes in the same category. To the best of our knowledge, there is neither prior work that studies the common misconceptions in spatial visualization sketching, nor cognitive models that describe the process of this task. The closest available work in cognitive models for spatial ability focuses on how people solve multiple choice spatial visualization questions, i.e., when candidate solutions are provided [14, 11, 31]. These models do not cover the process of generating a spatial object from scratch, which is what sets spatial visualization sketching apart from the traditional spatial ability tests. Hence, our proposed model is unable to distinguish the errors by their causes. Future research conducting qualitative interviews with students to understand the reasons why an error occur could provide valuable insights towards identifying not only broad categories of erroneous answer, but also the causes behind various error categories. It would also be beneficial to create cognitive models to understand systematically the strategies students used to solve these problems. These information would be valuable in further developing other features that could distinguish errors according to their underlying cause, for example, by leveraging the temporal sequence of actions executed by the student leading to their error. Improving current models to include information about the most probable cause of an error would be beneficial in generating formative feedback that goes beyond providing information about the nature of the students' error, and integrates conceptual information to support students in addressing misconceptions.

The current training data for the model only involved 14 students, which is a relatively small sample. As such, the current model can be seen as a proof-of-concept for the feasibility of erroneous answer categorization. Applying the same approach to a larger population of students will be necessary to validate the stability of the model and ensure that there are no additional answer categories that may not have been included in our current dataset. Future studies can re-train and test the model on a larger population to confirm the existence of the answer categories identified within the current study. Since the training process of the model is simple, re-training the model based on another dataset would be straightforward.

Another next step for this research is to deploy the model in an online training platform and conduct user testing to examine the effectiveness and accuracy of the categorization and intervention. Last but not least, the method proposed in this study is designed to be flexible and be applied to other disciplines. Future work in other disciplines, such as evaluating circuit diagrams in Electrical Engineering, engine models in Mechanical Engineering, building plans in Architecture, and structural formula in Organic Chemistry, will

need to be conducted to evaluate the extent to which the proposed method generalizes to new topics.

7. CONCLUSION

In conclusion, this paper presents a clustering model as a solution to categorize erroneous answers in complex free-hand sketching questions in spatial visualization training. Eight abstract features were developed and proven to be effective in the categorization of erroneous answers, including percentages of various types of incorrect lines, number of mistake components, and metrics of the size of the components. The clustering model detected six answer categories based on the severity and scale of the mistakes. With these detected categories, an online training platform will be able to present customized and actionable formative feedback in real-time. Moreover, our approach suggested a new and comprehensive set of evaluation criteria to assess a sketch, which could potentially be generalized to other disciplines that require sketching practices.

8. REFERENCES

- [1] Y. Attali and D. Powers. Effect of immediate feedback and revision on psychometric properties of open-ended gre® subject test items. *ETS Research Report Series*, 2008(1):i-23, 2008.
- [2] H. Ault and A. Fraser. A comparison of manual vs. online grading for solid models. In *Proceedings of the 2013 ASEE Annual Conference, Atlanta, Georgia, June 23*, volume 26, 2013.
- [3] H.-D. Barke and T. Engida. Structural chemistry and spatial ability in different cultures. *Chemistry Education Research and Practice*, 2(3):227-239, 2001.
- [4] S. Basu, C. Jacobs, and L. Vanderwende. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1:391-402, 2013.
- [5] A. N. Bhat. *Sketchography-Automatic Grading of Map Sketches for Geography Education*. PhD thesis, 2017.
- [6] C. S. Carter, M. A. Larussa, and G. M. Bodner. A study of two measures of spatial ability as predictors of success in different levels of general chemistry. *Journal of research in science teaching*, 24(7):645-657, 1987.
- [7] C. Chandan, M. Deepika, S. Suraksha, and H. Mamatha. Identification and grading of freehand sketches using deep learning techniques. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1475-1480. IEEE, 2018.
- [8] M. Contero, F. Naya, P. Company, J. L. Saorin, and J. Conesa. Improving visualization skills in engineering education. *IEEE Computer Graphics and Applications*, 25(5):24-31, 2005.
- [9] R. T. Duesbury et al. Effect of type of practice in a computer-aided design environment in visualizing three-dimensional objects from two-dimensional orthographic projections. *Journal of Applied Psychology*, 81(3):249, 1996.
- [10] M. O. Dzikovska, J. D. Moore, N. Steinhäuser, G. Campbell, E. Farrow, and C. B. Callaway. Beetle ii: a system for tutoring and computational linguistics experimentation. In *Proceedings of the ACL 2010*

- System Demonstrations*, pages 13–18. Association for Computational Linguistics, 2010.
- [11] D. E. Egan. Testing based on understanding: Implications from studies of spatial ability. *Intelligence*, 3(1):1–15, 1979.
 - [12] H. B. Gerson, S. A. Sorby, A. Wysocki, and B. J. Baartmans. The development and assessment of multimedia software for improving 3-d spatial visualization skills. *Computer Applications in Engineering Education*, 9(2):105–113, 2001.
 - [13] B. J. Gimmetstad. Gender differences in spatial visualization and predictors of success in an engineering design course. In *Proceedings of the National Conference on Women in Mathematics and the Sciences*, number 801, pages 133–136, 1990.
 - [14] J. Gluck and S. Fitting. Spatial strategy selection: Interesting incremental information. *International Journal of Testing*, 3(3):293–308, 2003.
 - [15] C.-C. Han, C.-H. Chou, and C.-S. Wu. An interactive grading and learning system for chinese calligraphy. *Machine Vision and Applications*, 19(1):43–55, 2008.
 - [16] M. Hegarty, M. Keehner, C. Cohen, D. R. Montello, and Y. Lipka. The role of spatial cognition in medicine: Applications for selecting and training professionals. *Applied spatial cognition*, pages 285–315, 2007.
 - [17] Y. Kali and N. Orion. Spatial abilities of high-school students in the perception of geologic structures. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, 33(4):369–391, 1996.
 - [18] S. Keshavabhotla, B. Williford, S. Kumar, E. Hilton, P. Taele, W. Li, J. Linsey, and T. Hammond. Conquering the cube: learning to sketch primitives in perspective with an intelligent tutoring system. In *Proceedings of the Symposium on Sketch-Based Interfaces and Modeling*, pages 1–11, 2017.
 - [19] S. Kirstukas. A preliminary scheme for automated grading and instantaneous feedback of 3d solid models. In *Proceedings of the midyear conference of engineering design graphics division of the ASEE*, pages 53–58, 2013.
 - [20] S. J. Kirstukas. Development and evaluation of a computer program to assess student cad models. In *Proceedings of ASEE’s 123rd Annual Conference and Exposition, New Orleans, LA*, page 26781, 2016.
 - [21] C. Leacock and M. Chodorow. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4):389–405, 2003.
 - [22] J. M. Leake and J. L. Borgerson. *Engineering design graphics: sketching, modeling, and visualization*. J Wiley & Sons, 2013.
 - [23] R. Lehming, J. Gawalt, S. Cohen, and R. Bell. Women, minorities, and persons with disabilities in science and engineering: 2013. *National Science Foundation, Arlington, VA, USA, Rep*, pages 13–304, 2013.
 - [24] O. L. Liu, C. Brew, J. Blackmore, L. Gerard, J. Madhok, and M. C. Linn. Automated scoring of constructed-response science items: Prospects and obstacles. *Educational Measurement: Issues and Practice*, 33(2):19–28, 2014.
 - [25] D. Lubinski. Spatial ability and stem: A sleeping giant for talent identification and development. *Personality and Individual Differences*, 49(4):344–351, 2010.
 - [26] J. D. Merrel, P. F. Cirillo, P. M. Schwartz, and J. Webb. Multiple-choice testing using immediate feedback-assessment technique (if at®) forms: Second-chance guessing vs. second-chance learning? 2015.
 - [27] N. S. Newcombe and A. Frick. Early education for spatial intelligence: Why, what, and how. *Mind, Brain, and Education*, 4(3):102–111, 2010.
 - [28] D. J. Nicol and D. Macfarlane-Dick. Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in higher education*, 31(2):199–218, 2006.
 - [29] R. Nielsen, W. Ward, and J. H. Martin. Classification errors in a domain-independent assessment system. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, pages 10–18, 2008.
 - [30] N. Orion, D. Ben-Chaim, and Y. Kali. Relationship between earth-science education and spatial visualization. *Journal of Geoscience Education*, 45(2):129–132, 1997.
 - [31] S. E. Poltrock and P. Brown. Individual differences in visual imagery and spatial ability. *Intelligence*, 8(2):93–138, 1984.
 - [32] J. R. Pribyl and G. M. Bodner. Spatial ability and its role in organic chemistry: A study of four organic courses. *Journal of research in science teaching*, 24(3):229–240, 1987.
 - [33] S. Pulman and J. Sukkarieh. Automatic short answer marking. In *Proceedings of the second workshop on Building Educational Applications Using NLP*, pages 9–16, 2005.
 - [34] K. Rochford. Spatial learning disabilities and underachievement among university anatomy students. *Medical education*, 19(1):13–26, 1985.
 - [35] J. Roorda. Visual perception, spatial visualization and engineering drawing. *Engineering Design Graphics Journal*, 58(2):12–21, 1994.
 - [36] V. J. Shute. Focus on formative feedback. *Review of educational research*, 78(1):153–189, 2008.
 - [37] I. M. Smith. *Spatial ability: Its educational and social significance*. RR Knapp, 1964.
 - [38] S. A. Sorby. Developing 3-d spatial visualization skills. *Engineering Design Graphics Journal*, 63(2), 2009.
 - [39] S. A. Sorby. Educational research in developing 3-d spatial skills for engineering students. *International Journal of Science Education*, 31(3):459–480, 2009.
 - [40] P. Taele and T. Hammond. Boponoto: An intelligent sketch education application for learning zhuyin phonetic script. In *DMS*, pages 101–107, 2015.
 - [41] D. H. Uttal and C. A. Cohen. Spatial thinking and stem education: When, why, and how? In *Psychology of learning and motivation*, volume 57, pages 147–181. Elsevier, 2012.
 - [42] D. H. Uttal, N. G. Meadow, E. Tipton, L. L. Hand, A. R. Alden, C. Warren, and N. S. Newcombe. The malleability of spatial skills: A meta-analysis of training studies. *Psychological bulletin*, 139(2):352,

2013.

- [43] S. Valentine, R. Lara-Garduno, J. Linsey, and T. Hammond. Mechanix: A sketch-based tutoring system that automatically corrects hand-drawn statics homework. In *The impact of pen and touch technology on education*, pages 91–103. Springer, 2015.
- [44] N. L. Veurink and A. Hamlin. Spatial visualization skills: Impact on confidence in an engineering curriculum. In *American Society for Engineering Education*. American Society for Engineering Education, 2011.
- [45] J. Wai, D. Lubinski, and C. P. Benbow. Spatial ability for stem domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of educational Psychology*, 101(4):817, 2009.
- [46] D. B. West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.
- [47] Z. Xiao, Y. Yao, C.-H. Yen, S. Dey, H. Wauck, J. M. Leake, B. Woodard, A. Wolters, and W.-T. Fu. A scalable online platform for evaluating and training visuospatial skills of engineering students. In *2017 ASEE Annual Conference & Exposition. ASEE Conferences, Columbus, Ohio*. <https://peer.asee.org/27509>, 2017.
- [48] Y. Zhang, R. Shah, and M. Chi. Deep learning+ student modeling+ clustering: A recipe for effective automatic short answer grading. *International Educational Data Mining Society*, 2016.

9. ACKNOWLEDGMENTS

The authors would like to thank Brian Woodard, Ziang Xiao, and the rest of the SIIP project team at the University of Illinois, Urbana-Champaign for the joint effort in developing the spatial visualization online training platform and offering the Fall 2019 "Spatial Visualization" course.

Getting too personal(ized): The importance of feature choice in online adaptive algorithms

ZhaoBin Li
Carleton College
liz2@carleton.edu

Luna Yee
Carleton College
yeec@carleton.edu

Nathaniel Sauerberg
Carleton College
sauerbergn@carleton.edu

Irene Sakson
Carleton College
saksoni@carleton.edu

Joseph Jay Williams
University of Toronto
williams@cs.toronto.edu

Anna N. Rafferty
Carleton College
arafferty@carleton.edu

ABSTRACT

Digital educational technologies offer the potential to customize students' experiences and learn what works for which students, enhancing the technology as more students interact with it. We consider whether and when attempting to discover how to personalize has a cost, such as if the adaptation to personal information can delay the adoption of policies that benefit all students. We explore these issues in the context of using multi-armed bandit (MAB) algorithms to learn a policy for what version of an educational technology to present to each student, varying the relation between student characteristics and outcomes and also whether the algorithm is aware of these characteristics. Through simulations, we demonstrate that the inclusion of student characteristics for personalization can be beneficial when those characteristics are needed to learn the optimal action. In other scenarios, this inclusion decreases performance and increases variation in student experiences. Moreover, including unneeded student characteristics can systematically disadvantage students with less common values for these characteristics. Our simulations do however suggest that real-time personalization will be helpful in particular real-world scenarios, and we illustrate this through case studies using existing experimental results in ASSISTments [23]. Overall, our simulations show that adaptive personalization in educational technologies can be a double-edged sword: real-time adaptation improves student experiences in some contexts, but the slower adaptation and increased variability mean that a more personalized model is not always beneficial.

Keywords: multi-armed bandits, personalization, educational technologies, online adaptive algorithms, simulation

1. INTRODUCTION

Within educational technologies, there are a myriad of ways to design instructional components such as hints or expla-

nations. Research in education and the learning sciences provides some insight into how to design these resources (e.g., [25, 3]). However, there is often uncertainty about which version of a resource will be most effective in a particular context, and effectiveness may vary based on students' characteristics, such as prior knowledge or motivation.

Randomized experiments are one way to compare multiple versions of a technology, but such experiments impose a delay between collecting required evidence and using that evidence to improve student experiences. Recently, multi-armed bandit (MAB) algorithms have been proposed to improve technologies in real time: each student is assigned to one version of the technology, and the algorithm observes the student's learning outcome [18, 28]. Each subsequent student is more likely to be assigned to a version of the technology that has been more effective for previous students, as the algorithm discovers what is effective. Such algorithms maintain uncertainty as they learn, balancing exploring to learn more about what works with exploiting the observed results from previous students. Typical MAB algorithms do not take into account student characteristics and thus can only identify which version of a technology is better for students on average, but contextual MAB algorithms can personalize which version to assign to each student, potentially increasing the number of students who are directed to versions that are most helpful for them individually [24].

While deploying contextual MAB algorithms could improve student experiences, it raises two potential issues. First, instructional designers must decide which student characteristics will be considered for personalization. For instance, more concrete examples might be more helpful for students with lower prior knowledge, while more abstract examples could be more helpful for students with higher prior knowledge. This relationship could only be learned if the algorithm has 'prior knowledge' as a feature of each student. Should the algorithm also consider which prerequisite course was taken when selecting an example, or is prior knowledge sufficient? Designers are unlikely to be certain which characteristics influence effectiveness, but the choice of characteristics will influence the performance of the algorithm. Excluding characteristics that do impact effectiveness could decrease the positive impact on students, but including extraneous characteristics that do *not* impact effectiveness could

Zhaobin Li, Luna Yee, Nathaniel Sauerberg, Irene Sakson, Joseph Jay Williams and Anna Rafferty "Getting too personal(ized): The importance of feature choice in online adaptive algorithms" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 159 - 170

also decrease this impact. In the latter case, the system might have to do more exploration to learn how the effectiveness of instruction differs along each extraneous characteristic, and so direct a greater number of students to less effective versions.

The second issue raised by online adaptive algorithms is whether the constantly adapting system will benefit certain groups of students more than others. Since contextual MAB algorithms learn by observing how the consequences of their choices are related to feature values, students whose characteristics are less common may be more likely to interact with the algorithm when it has limited information about what is most effective for that type of student. This could exacerbate differences in outcomes between subgroups of students. Yet, such algorithms could also have an equalizing effect for students with less common characteristics: students have the potential to experience a version of the technology that is most appropriate for them, even when this version is not the most appropriate for a typical student.

In this paper, we use simulations to explore these issues and their consequences for student experiences in adaptive educational technologies which use MAB algorithms. We focus on three common types of models for how student characteristics are related to outcomes: a *baseline* model in which student characteristics do not impact the effectiveness of different versions of the technology; a *universal optimal action* model, in which student characteristics impact effectiveness but the same version is most effective for all students; and a *personalized optimal action* model, in which student characteristics impact which version leads to the best outcomes for a given student.

We show that including the potential for personalization significantly degrades student outcomes except in the *personalized optimal action* model, where this information is necessary to encode the best policy. While the cost of including more characteristics for personalization is relatively modest, including these characteristics leads to greater variation: the algorithm is less consistent in learning which versions are best overall, and students may be systematically treated differently based on characteristics that do not influence their outcomes. This increased variance is worsened when student characteristics are not uniformly distributed, with some characteristics being more common than others. We use experimental data to show the potential benefits of personalization and add nuance to the prior simulation results by demonstrating how personalization can benefit not only students in a minority group but also all groups of students. We end by discussing the consequences of these results for integrating adaptive components into existing educational technologies.

2. RELATED WORK

A wide array of work has focused on using MAB and contextual MAB algorithms for optimization, including applications in advertising and recommendations (e.g., [17]), crowdsourcing (e.g., [14]), and designing experiments and clinical trials (e.g., [26]). Within educational technologies, MAB algorithms have been primarily used in two ways. Some work has used these algorithms to select problems that are of an appropriate difficulty level for a particular student [8,

16, 22]; unlike our work, these applications typically combine learned profiles about students with a second source of knowledge, such as prerequisite structure. We focus on a second proposed usage of MAB algorithms in education: assigning students to a particular version of a technology. For example, non-contextual MAB algorithms have been used to choose among crowdsourced explanations [27] and to explore an extremely large range of interface designs [19]. Some of this work has also considered the implications of collecting experimental data via MAB algorithms on measurement and inference [18, 20], showing systematic biases that can impair the drawing of conclusions about the conditions. Only a limited amount of work has applied contextual MAB algorithms to personalize which versions of a technology a student experiences (e.g., [24], but focused primarily on measurement). We build on this body of work by considering the performance implications of several common scenarios for how student characteristics, versions of an educational technology, and outcomes are related. Additionally, by specifically examining some scenarios in which student characteristics are unevenly distributed, we raise issues about personalization for minority groups of students.

There is a great deal of theory-based literature on both standard and contextual MAB algorithms related to quantifying performance, especially in terms of asymptotically bounding growth in cumulative regret (the amount that the expected reward from choosing an optimal action outpaces reward from the actually chosen actions). The optimal worst-case bound on regret growth is logarithmic [4]. Furthermore, the inclusion of contextual variables increases cumulative regret at least linearly; for Thompson sampling, which we use in our simulations, the regret bounds grow quadratically in the number of contextual variables [2]. We use simulations to consider non-asymptotic settings and focus on areas less explored theoretically, like impacts on individual groups of students and variability in performance.

In this paper, we are particularly concerned with how outcomes differ among different groups of students. One of the promises of educational technologies is to boost all students' outcomes to the level that can be achieved by individualized tutoring [9], and online adaptive algorithms may make it easier to develop such systems. Yet, the broader machine learning community has recently highlighted how automated systems can learn or exacerbate existing inequalities (see, e.g., [12] for an overview). Within educational data mining, there have been mixed results when the fairness of different models has been explored, and this variation has often been correlated with the diversity of the training data: [13] demonstrated that a model trained on a large and diverse dataset performed similarly well for predicting on-time graduation for students in different demographic groups, while [11] found disparities across genders in predicting course dropout, often associated with gender imbalances in the training data. This raises the issue of how to best use educational data mining in ways that promote equity across students. Within the MAB literature specifically, there has been limited discussion of fairness (e.g., [15]), although [21] show that a particular technical definition of data diversity can lead to fairer outcomes. Like in our work, [21] shows cases where the presence or absence of a majority group can help or harm minority group outcomes. Our work con-

siders scenarios specific to education, demonstrating that the particular scenario in [21] can be generalized considerably, and more precisely characterizes the circumstances in which including personal characteristics increases equity versus where doing so may lead to systematically poorer experiences for students in a minority group.

3. CONTEXTUAL MAB ALGORITHMS

We treat the problem of determining what version of an educational technology will be most effective for a student as a MAB problem. In such problems, a system must repeatedly choose among several actions, a_1, \dots, a_K . The system initially does not know which action is likely to be the most effective, but after each action choice, the system receives feedback in the form of a stochastic reward $r^{(t)}$.

There are a variety of MAB algorithms for choosing actions. We focus on Thompson sampling [1], which is a regret-minimizing algorithm that exhibits logarithmic regret growth. Thompson sampling maintains for each action a distribution over reward values. This distribution is updated after each action choice and represents the posterior distribution over reward values given the observed data. At each timestep, the algorithm samples from the posterior distribution over rewards for each action, and then chooses the action with the highest sampled value. While Thompson sampling is also applicable to real-valued rewards, many educational outcomes are binary, such as whether a student completes a homework assignment or answers a question correctly. Thus, we focus on these binary rewards in this paper, using a Beta prior distribution to enable simple conjugate updates after each choice.

In our setting in which we choose versions of an educational technology for each student, the actions are the different versions of the technology, and the reward is the student outcome. For example, imagine a student interacting with a system to do her math homework. The system might choose between two actions when the student asks for a hint: (a) show a fully worked example, versus (b) provide the first step of the problem as a hint and ask the student to identify an appropriate second step. The student outcome could be whether or not she completes the homework assignment.

In a traditional MAB problem, the reward distribution is fixed given the action choice. However, in the situation above, the reward may be dependent on the characteristics of the student. For instance, a student who has stronger proficiency in the prerequisite skills may be more prepared to identify what to do next in the problem, while a student with weaker proficiency may not be able to identify what to do next. A contextual MAB algorithm incorporates such student characteristics as features into its action choices.

For parametric contextual MAB algorithms, the features must be predetermined, including whether interactions between features is permitted. We adopt a contextual Thompson sampling approach that uses regularized Bayesian logistic regression to approximate the distribution of rewards given the features [2, 7]. The algorithm learns a distribution over the feature weights as coefficients using a Gaussian posterior approximation. To make each new action choice, the algorithm computes a reward value for each action by

sampling each weight independently. The chosen action is the action with the highest sampled reward value. Updates may occur after each action or in batches to decrease computational costs; because the feature vectors that we consider are relatively small, we update after each action.

4. IMPORTANCE OF FEATURE CHOICE

When using a contextual MAB algorithm to personalize student experiences in an educational technology, the system designer must choose which student characteristics to include as variables for personalization. The designer is very unlikely to know with certainty which student features are truly relevant and will actually impact student outcomes. One could include every possible relevant feature, knowing that while the algorithm can learn that an included feature is not relevant, it cannot learn that a non-included feature is in fact relevant. However, asymptotic growth rates for regret are quadratic in the number of features [2], meaning that as more features are included, the algorithm will tend to take longer to learn. Designers thus must balance the desire to include all features that influence outcomes with the knowledge that extraneous features could hurt performance.

To better understand how student outcomes are impacted by the choice of features for personalization, we systematically explore the inclusion of both relevant and non-relevant features in a contextual MAB algorithm and examine the impact on student outcomes and on the rate of assigning students to their personally optimal version of the technology. For these simulations, we assume that features are uncorrelated and that their values are chosen uniformly at random for each student, i.e., the probability of observing any particular combination of features is the same as observing any other combination of features.

4.1 Methods

4.1.1 Representing student features

We focus on binary student features and thus feature values implicitly group students. For example, some CS classes may have two different prerequisites, such as a discrete math course taught by the CS department or a similar one taught by the math department. Students who have taken the CS version will all have the same value for the prerequisite feature, while those who take the math one will have the other.¹

4.1.2 Outcome-generating models

The outcome-generating model describes the *true* relationship between student characteristics (feature values), the actions of assigning students to different versions of a technology, and the outcomes of student learning. We focus on scenarios in which two actions, such as choosing between concrete versus abstract explanations, affect the outcomes for two groups of students, such as those with math versus CS prerequisite as aforementioned.

In each of the models, we generate the true reward probability for a student with particular features using logistic regression, with a separate logistic regression equation for

¹In both the MAB algorithms and the outcome-generating models, feature values are represented using dummy variables.

Relevant Feature: Action Number:	F=0		F=1	
	A1	A2	A1	A2
Baseline	0.4	0.6	0.4	0.6
Universal optimal action (1)	0.4	0.6	0.6	0.8
Universal optimal action (2)	0.4	0.6	0.4	0.8
Universal optimal action (3)	0.4	0.6	0.5	0.7
Universal optimal action (4)	0.4	0.6	0.8	0.9
Personalized optimal action	0.4	0.6	0.6	0.4

Table 1: Reward probabilities for each combination of actions (A1 and A2) and values of the relevant feature (F=0 and F=1) in the simulations. The optimal action, shown in bold, is the same (A2) for both feature values, except for the personalized optimal action model.

each action. Given a feature vector $x^{(j)}$ for student j , the reward probabilities are generated according to:

$$P_{action=k}(\text{reward} = 1 \mid x^{(j)}) = \text{sigmoid}(b_{0,k} + \sum_{i=1}^n b_{i,k}x_i),$$

where b_k is the coefficient vector for action k and has intercept $b_{0,k}$. For our simulations, the coefficients for the feature values were zero for any feature past the first feature, meaning that a maximum of one student feature impacts the outcomes but more features may still be observed. By varying the coefficients for the intercept ($b_{0,k}$) and the first feature, we produced three models for the relationship among student characteristics (i.e., features or feature values), action choices, and outcomes (see Table 1):

- *Baseline*: Student features have no impact on outcomes.
- *Universal optimal action*: Student features have an impact on outcomes, but not the optimal action—the best version of the technology is the same regardless of features.
- *Personalized optimal action*: Student features impact outcomes, meaning that the optimal action differs based on features—some students are better off experiencing Version A of the technology while for others Version B.

For the baseline model, the coefficients of the actions vary only for the intercept in order to control the effect of each action when student features are ignored. For the universal optimal action model, we included four variations to capture different educationally meaningful scenarios. For instance, universal optimal action (1) reflects a case in which differences in prior knowledge minimally interact with the impact of different versions of a technological intervention, while (2) reflects a student characteristic magnifying the effectiveness of an intervention.

4.1.3 Simulation parameters

We varied three factors across the simulations: the outcome-generating model; the MAB algorithm (contextual or non-contextual); and the number of student features. For all simulations, we considered three horizons: classrooms of 50, 250, and 1000 students. Multiple horizons illustrate the behavior of the algorithm at different time points and can guide decisions for incorporating adaptive algorithms based on the number of students who are expected to interact with the system. Each simulation was repeated 1000 times.

For the non-contextual Thompson sampling, parameters for a Beta distribution per action are learned independent of student features. For the contextual algorithm, we specify the weights of the student features as model coefficients. All simulations included at least one student feature regardless of the outcome-generating models.

To model the fact that curriculum designers may not know which student characteristics really matter, we included simulations where the observed features were a superset of those that actually impacted outcomes. Specifically, we considered models with a total of 1, 2, 3, 5, 7, 8, and 10 features. Therefore, for the non-baseline scenarios, the proportion of included features that impacted outcomes varied from 100% to only 10%. Since our contextual features are binary, we include indicator variables for each of the two values, and learn a separate weight for each indicator variable.²

4.2 Results

First we focused on analyzing the performance of contextual and non-contextual MAB algorithms for the three outcome-generating models across 1 to 10 student features (i.e., contextual variables). Using an analysis of covariance (ANCOVA), we compared the two MAB algorithms' performance with respect to the proportion of optimal actions for 250 students across 1000 trials, treating the number of contextual variables as a covariate.

Baseline: When student features do not influence outcomes, we see that as expected, the non-contextual bandit outperforms the contextual bandit (Table 2): average performance per student for the final 50 out of 1000 students using the contextual algorithm is similar to that of the first 250 students using the non-contextual algorithm (Figure 2). As the number of student features increases, the contextual MAB chooses a lower proportion of optimal actions for the first 250 students (Figure 1a), but the effect is relatively small especially when considering the impact on actual reward ($t(13996) = -10.880$, $p < 0.001$, $b = -0.006$, 95% CI = $[-0.007, -0.005]$). At longer horizons, the number of student features has less of an impact on overall average reward (Figure 2), which we discuss more below.

Universal optimal action: When outcomes are dependent on student features, the contextual MAB algorithm can learn a more accurate model than the non-contextual algorithm. However, when this more accurate model is not needed for optimal action choices, learning the more accurate model does not improve action choices: the non-contextual bandit outperforms the contextual bandits in all four scenarios (Table 2; see Figure 1b for scenario 1). While each scenario might arise due to different educational conditions, they are all very similar in how they appear to the non-contextual bandit algorithm. The non-contextual bandit sees the two groups of students as identical, leading the overall performance to be the average for each group. These changes in the average effectiveness of each intervention impact the algorithm's performance but do not necessarily degrade that performance; instead, the impact is dependent

²In pilot simulations, this encoding led to better performance than if only a single coefficient was learned for each feature, and corrected asymmetries in performance for students who had different values of the feature.

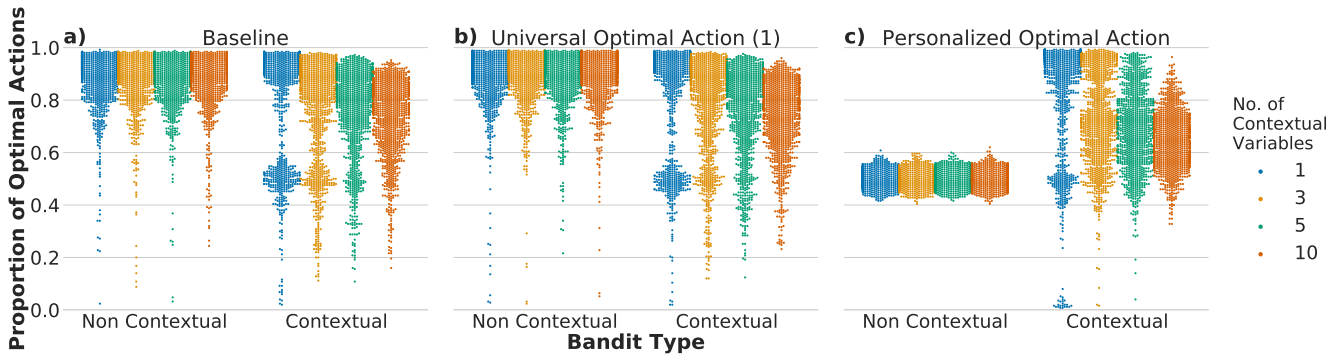


Figure 1: Swarm plots for the proportion of optimal actions for the two bandit types. Each point represents results from one trial with 250 students. For the universal optimal action, all scenarios show similar results; hence only scenario (1) is shown. The bimodality of the contextual bandits, especially at low numbers of contextual variables, highlights the potential risks of personalization.

	Superior bandit	$ b $	95% CI	$F(1, 13996)$	p	Cohen's d
Baseline	Non Contextual	0.098	[0.089, 0.108]	2678.0	< .001	0.871
Universal optimal action (1)	Non Contextual	0.088	[0.079, 0.097]	2750.0	< .001	0.880
Universal optimal action (2)	Non Contextual	0.078	[0.072, 0.085]	3853.0	< .001	1.042
Universal optimal action (3)	Non Contextual	0.101	[0.092, 0.11]	2891.0	< .001	0.904
Universal optimal action (4)	Non Contextual	0.074	[0.063, 0.085]	1865.0	< .001	0.725
Personalized optimal action	Contextual	0.295	[0.287, 0.302]	10816.0	< .001	1.677

Table 2: Inferential statistics for proportion of optimal actions for the two bandit types across all outcome-generating models, simulated for 1000 trials of 250 students each. b represents the coefficient of improvement of results for the superior bandit after controlling for the number of contextual variables.

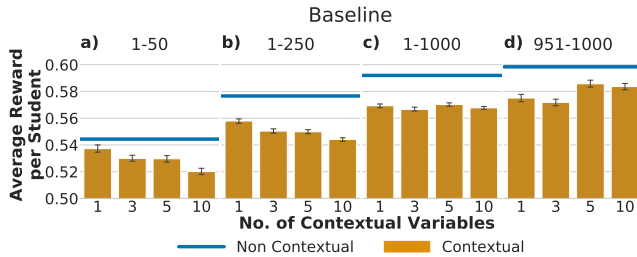


Figure 2: Average reward per student across 1–10 contextual variables for the two bandit types in the baseline model. In this model, the maximum expected reward is 0.6, and the expected reward for uniform random assignment is 0.5. Error bars represent 1 standard error.

on how similar the two interventions are in their expected outcomes and how close those expected outcomes are to 0.5, where there is the most variance.

Personalized optimal action: When the best policy for individual students depends on their features, the contextual bandit significantly outperforms the non-contextual bandit (Table 2). When only one student feature is included, the contextual MAB algorithm chooses the optimal action almost 70% of the time for the first 50 students; this increases to almost 90% for the final 50 of the total 250. Including extra student features decreases performance - if ten features are included and only one impacts the policy, the overall

proportion of optimal actions falls to about 65%. Yet, this is still an improvement over the non-contextual algorithm (Figure 1c). These results suggest that even if a relatively small number of students will interact with the system and one is uncertain about which of a (limited) set of features will impact results, including those features will on average have a positive impact on student outcomes if one is confident that the best version of the system for an individual student varies based on one of those features.

Variability across simulations: Examining variability across simulations provides insight into how likely actual deployments of these algorithms are to reflect their average performance. Across all models, the contextual MAB algorithm exhibited greater variability in performance than the non-contextual MAB algorithm (Figure 1). Surprisingly, increasing the number of student features leads to lower variance for the contextual MAB algorithm. With small numbers of student features, there is often a concentration of simulations with lower achieved outcomes, resulting in bimodal distributions (Figure 1). The bimodality emerges because the algorithm can adapt more quickly, making it somewhat more vulnerable to underestimating parameter values based on a few samples with unexpected low rewards. Because the parameter estimates influence future action choices, data to correct these underestimates may not be collected quickly enough (as has been documented for non-contextual bandits in, e.g., [10]). In contrast, increasing the number of student features increases variation near the mean but eliminates the bimodality (Figure 1) since the algorithm performs more exploration to learn the larger number of parameters.

This makes it less likely to collect data that lead to erroneous conclusions about the effectiveness of actions. Errors in the parameter values are more likely to be corrected because they are unlikely to lead to the same choices for all student features, hence creating more variability in action choice for students with a specific value of a single feature. Indeed, the simulation results support these interpretations: for the baseline and universal optimal action models in a 1000-student classroom (see Figure 2 for baseline), average reward for the first 250 students is lower but reward for the final 50 students is higher as the number of contextual variables increases. There is thus a trade-off between expected outcomes and variability: the ability of the contextual MAB algorithm to adapt more quickly when it has fewer features to learn comes at the cost of it being less able to correct for wrong conclusions from small amounts of data.

Variability in policies across students: As noted above, the extra parameters learned by the contextual MAB algorithm lead to the potential for greater variability in action choices within a single simulation. This can systematically affect groups of students when the algorithm attaches spurious relevance to a feature that does not actually impact outcomes. We can see this pattern by examining differences in action probabilities for students who differ only by characteristics that do not impact outcomes: that is, considering all students who have the same value for the first feature, how does the probability of choosing a particular action change based on their different values for the other features? As the number of contextual variables increases, the average maximum difference in action choice probability between such students also increases from 18–25% when two student features are included in the model to over 90% when ten features are included in the model after running through 250 students. This occurs both based on the greater expressivity of the model with more student features and the fact that the model with more student features is likely still learning about the impact of each of these features. This raises potential concerns about inequity: students who should be treated identically by the system may instead be treated systematically differently, based on features that do not impact how they learn.

5. IMPACT OF UNEVEN DISTRIBUTION OF STUDENT CHARACTERISTICS

The results of the previous simulations demonstrate that in situations where student characteristics (features) impact the outcome of different educational interventions, a contextual MAB algorithm only provides an improvement over a non-contextual algorithm when knowledge about the characteristic is necessary for choosing the best action. These simulations provided insight into how performance is impacted by different patterns of relationships between student characteristics and outcomes, with the assumption that those characteristics were uniformly distributed. However, in reality, some characteristics are likely to be more common than others. For example, when optimizing which hint to give to students who answer a question incorrectly, the algorithm is more likely to encounter a student with lower prior knowledge than one with higher prior knowledge. Thus we now relax this assumption and explore how changing the distribution of student characteristics impacts student outcomes for both types of MAB algorithms. In these simulations, we

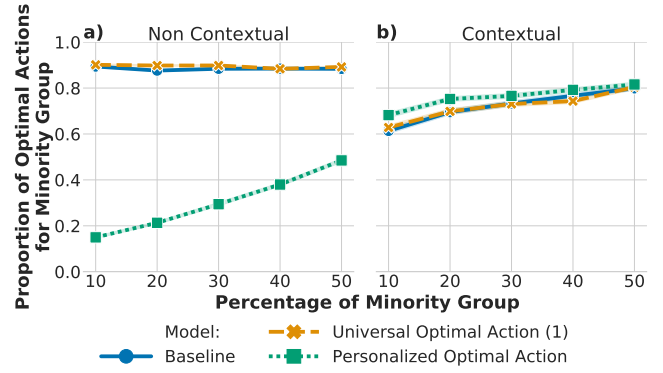


Figure 3: Proportion of optimal actions for minority groups with sizes of 10%–50% for the two bandit types across the three outcome-generating models, limited to one contextual variable. Standard errors, represented by the translucent bands, are negligible.

examined not only overall outcomes, but also outcomes for different groups of students. Attention to group-specific outcomes is vital for identifying inequitable impacts of adaptive algorithms.

5.1 Methods

Similar to the first set of simulations, we compared non-contextual and contextual MAB implementations that used Thompson sampling across the same three horizons of 50, 250, and 1000 students, with a focus on 250; we repeated each simulation 1000 times. These simulations include a new independent variable: the proportion of students in each group. Specifically, for each simulated student, we varied the probability of the student being in the minority group (i.e., having a value of one for the first student characteristic) from 10% to 50%, using 10% increments. In addition to analyzing performance across all students, we examined performance for both the minority and majority groups separately. We also examined the *balanced success rate*, defined as the simple average of the group-specific performances [5]. Balanced success rate provides a way of examining performance that treats each group as equally important, even though one group may have more students than another.

5.2 Results

As in the previous analysis, we used an ANCOVA to compare the performance for the two bandit types in terms of the proportion of optimal actions, but this time treating the percentage of the minority group as a covariate.

One student characteristic: With one student characteristic, the contextual MAB algorithm’s performance for the minority group decreases as the size of the minority group becomes smaller, across all outcome-generating models (Figure 3b and Figure 4; $t(59996) = -33.962$, $p < 0.001$, $b = -0.427$, 95% CI = $[-0.452, -0.402]$). This leads the contextual MAB algorithm to have a lower balanced success rate for smaller minority groups. However, overall performance across all students is slightly better since so many more students are in the majority group (Figure 4; $t(59996) = 16.633$, $p < 0.001$, $b = 0.126$, 95% CI = $[0.111, 0.141]$). In other

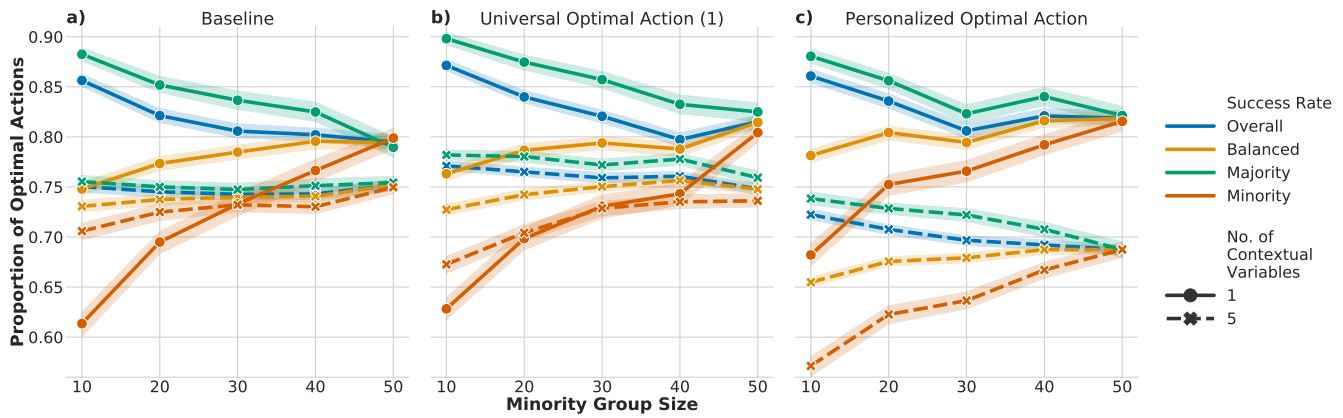


Figure 4: Comparing the proportion of optimal actions of the contextual bandit between 1 and 5 student features (i.e., contextual variables) for the majority and minority groups, as well as their balanced and overall averages, across minority group sizes of 10%–50%. Standard errors, represented by the translucent bands, are negligible.

words, decreasing the minority group size hurts the minority group more than it helps the majority group on a per-student basis; but replacing students from the minority group, who are assigned worse conditions, with students from the majority group, who are assigned better conditions, increases overall reward.

This pattern of results occurs because the contextual MAB has more uncertainty about the impact of the particular value of the student characteristic that appeared fewer times: in the least balanced case, we expect the minority group to be seen only 25 times on average given a horizon of 250 students. Hence, providing a model with the potential to personalize for a minority group is a calculated risk - although the extra expressivity is likely intended to improve experiences for all groups of students, it can negatively impact minority groups, with a larger negative impact for smaller minority groups.

In contrast, the non-contextual MAB algorithm is relatively unaffected by the changing distribution of student characteristics in both the baseline ($t(9996) = 0.497$, $p = 0.619$) and universal optimal action scenarios ($t(39996) = 1.506$, $p = 0.132$), as shown by Figure 3a. The changing distribution of student characteristics changes the expected rate of obtained reward from each action, but the changes are small enough that they have little impact on the algorithm’s ability to choose optimal actions.

However, for the personalized optimal action model, the size of the minority group *does* have a large impact on individual student outcomes for the non-contextual MAB algorithm: when the minority group is small, the algorithm learns to choose the action that is best for the majority and worst for the minority, resulting in the optimal action being chosen only 15% of the time for the minority group, within a horizon of 250 students (Figure 3a). When the two groups are of equal size, the algorithm has no systematic information that shows one action as consistently better or worse than the other; thus on average, it chooses the optimal action about 50% of the time for both groups.

Additional student characteristics for the contextual MAB algorithm: When the number of student characteristics increases, the impact on the minority and majority groups differs for the baseline and universal optimal action models compared to the personalized optimal action (Figure 4). In the two former models, the impact on balanced success rate is generally small: as the number of student characteristics increases from one to five, balanced success decreases no more than 8%, except by 11% in universal optimal action (4); for most of these models, the decrease is even smaller when the minority group is smaller. In these models, the algorithm’s performance for small minority groups is improved with more student characteristics, while performance for majority groups decreases. For example, in the baseline scenario with 10% of students in the minority group, the algorithm chooses the optimal action for 71% of the minority group when there are five student characteristics, compared to 61% of these students when there is only one student characteristic. More student characteristics leads to more exploration with the initial students, and thus the algorithm is less likely to systematically execute a bad policy for the minority group based on a small number of initial samples.

For the personalized optimal action scenario, increasing the number of student characteristics from one to five decreases performance for both minority and majority groups by about 15% regardless of the size of the minority group, uniformly lowering balanced success rate. Due to the extra exploration caused by the extraneous student characteristics, the algorithm is slower to exploit the actual relationship between the relevant student characteristic and the action choice, without differential impact based on minority group size.

6. REAL-WORLD EXPERIMENTS

The first two sets of simulations can guide system designers when making decisions about personalizing based on student features. However, they have some limitations: while they considered a relatively large space of possibilities for how outcomes relate to student features, they focused on showing a general variety of cases rather than on specific cases that might be most common or of particular interest

	Problem Set	Students	Q1 Size	Q2 Size	Q3 Size	Q4 Size
Uneven Student Distribution	293151	320	113 (35.3%)	100 (31.3%)	69 (21.6%)	38 (11.9%)
Even Student Distribution	263057	129	33 (25.6%)	28 (21.7%)	34 (26.4%)	34 (26.4%)

Table 3: Student totals and group distributions in the original ASSISTments data [23] for the two problem sets of interest. Prior percent correct is discretized before removing students who have never answered incorrectly to experience the assigned condition, biasing group size towards the lower quartiles in the Uneven Student Distribution.

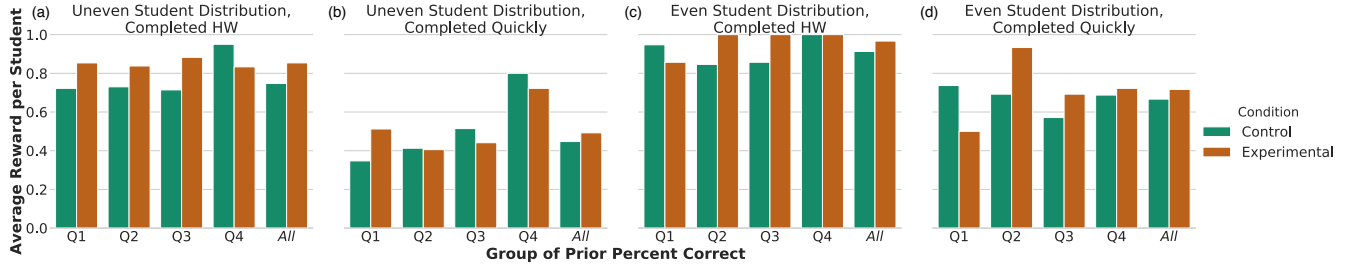


Figure 5: Original average reward per student in the ASSISTments data [23], across the four quartiles (Q1–Q4) of prior percent correct and their averages, for the two conditions in the experiments (control and experimental) illustrates our model parameters of real-world scenarios.

in education. To address this, we conducted several case studies of how MAB algorithms would have impacted actual experiments. We consider existing experimental data in which the optimal action would be personalized to see if the contextual MAB algorithms benefits students (as would be expected from our previous simulations) and also to demonstrate how factors from the previous simulations manifest in real-world scenarios.

The experiments were previously conducted within *ASSISTments*, an online learning system, and focused primarily on middle school math. We selected several experiments from [23] based on how student outcomes were related to their prior successes in the system as well as their assignment of either the control or experimental condition. Prior success in the system is a strong candidate to be a student feature for personalization: it is typically easily available and can serve as a proxy for prior knowledge, which has been shown to influence the success of different instructional strategies [25].

6.1 Methods

To model previously collected ASSISTments data in our MAB framework, we (1) transformed both the student characteristics and the student outcomes into discrete variables,³ and (2) resampled from the data to generate outcomes when the MAB algorithm assigned a condition.

For step (1), we first discretized students’ prior percent correct on problems within ASSISTments, the sole student feature that we included for personalization, into four quartiles: the 25% of students who began the homework assignment with the lowest prior percent correct (Q1), then those in the 26–50% range (Q2), and so on. The dataset contains some students who began the homework but were not assigned to

³MAB algorithms can handle non-categorical data, but we focus on the categorical case to mirror our prior simulations.

a condition. Since the experiments in [23] mainly manipulated students’ experiences (e.g. type of hint) when they answered a question incorrectly, students who have never answered incorrectly are not included in the experiment results (nor will the MAB algorithm make choices for them). However, they are included in the quartile cutoffs, which means that in the population of students with whom the MAB algorithm interacts, the number of students in each quartile may not be uniform.

We also chose and discretized the student outcome measures. These experiments included two different measures of student outcomes: whether each student completed the homework and the number of problems that each student answered in the homework. All experiments took place in the *SkillBuilder* interface, where students must answer three consecutive problems correctly to complete the homework. Completion of homework (denoted *Completed HW*) is already discrete and could easily be collected in real time; two of our simulations use this measure. However, it is relatively coarse, as the vast majority of students completed the homework. Thus, we also used a discretized version of the number of problems to completion (denoted *Completed Quickly*). If a student completes the homework, doing so in fewer problems is a better outcome than doing so in more problems. Outcomes were based on the median problem count for students who completed the homework. Students who completed the homework in the median number of problems or fewer had positive reward, while those who did not complete the homework or completed it more slowly had no reward. Though for practical use prior data would be needed to select an appropriate cut point, using a cut point based on collected data in our simulations measures the performance of students more closely.

For step (2), we simulate a MAB algorithm’s performance by repeatedly sampling students from the experiment. Within each trial, we fix the number of timesteps to the total num-

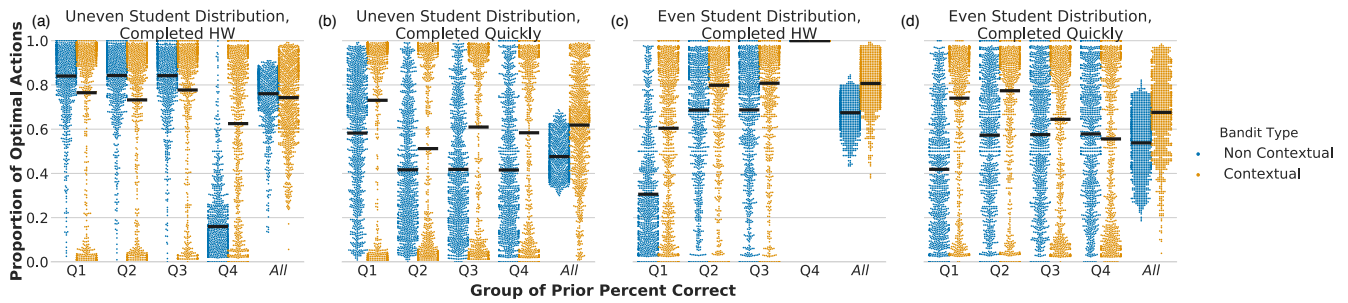


Figure 6: Swarm plots for the proportion of optimal actions for the two bandit types for each quartile of prior percent correct and their averages. Each point represents results from each of the 1000 trials per experiment and the solid black lines indicate the means of each swarm plots. Points for Q4 of Even Student Distribution, Completed HW are clustered at 1.0 because both actions are optimal. The extra information learned by the contextual bandit improves performance in most cases but the bimodality for some quartiles demonstrates the associated systematic risks.

ber of students in the original experiment. At each timestep, a random student is sampled, and the algorithm then selects a condition for that student. To compute the outcome, we sample from all outcomes for students in the experiment who were in the same quartile for prior percent correct and who experienced the same chosen condition. Each trial thus represents an experiment of the same size as the original, with the students drawn with replacement from the experimental data. We randomized each of the 1000 trials, though for each trial, we use the same student ordering for both types of MAB algorithms.

In our case studies, we focus on one problem set (#293151) where students are unevenly distributed across quartiles, with more lower-performing students (Q1), and one problem set (#263057) in which students are more evenly distributed across quartiles (see Table 6). With the two different outcome measures, this resulted in four simulation scenarios. We chose these problem sets because they had student outcomes that varied based on both condition and student quartile (see Figure 5).

6.2 Results

In all four settings, at least one quartile of students (out of Q1–Q4) was helped by the contextual MAB algorithm, and in three of the four settings, average outcomes across all students were improved by personalization.

Uneven Student Distribution, Completed HW: As shown in Figure 6a, in this scenario, students in Q4 were much more likely to experience their optimal condition with a contextual MAB algorithm. This occurs because the condition that is best for the average student is the one that is worse for Q4: the non-contextual MAB thus optimizes in a way that has a systematic, negative outcome for Q4 students. Conversely, the contextual MAB algorithm does not do as well as the non-contextual algorithm for students in Q1–Q3 because of the extra exploration needed to learn about more variables that are not necessary to help these students. Overall, this means that the contextual MAB algorithm had a slightly lower rate of choosing the optimal action than the non-contextual MAB. However, the difference is relatively small, and is even smaller in terms of average

reward: reward is reduced by less than 0.01 overall, while is increased for Q4 students by about 0.06. In this experiment, reward rates are high in general (greater than 70% for all conditions and quartiles). Thus with 320 students, small differences in condition assignment often are not reflected in large differences in outcomes. Q1–Q3 students have very similar outcomes across the two methods of condition assignment; Q4 has the greatest difference in success for one condition versus another, and thus the large increase in optimal condition assignment for these students does boost the average outcomes.

Uneven Student Distribution, Completed Quickly:

Using the Completed Quickly outcome measure with the same students, students in all quartiles were more likely to be assigned to the optimal condition when the contextual MAB algorithm was used (Figure 6b). This pattern occurs because the overall probability of a positive outcome is very similar across the two conditions when student quartiles are ignored (shown by *All* in Figure 5b), making it difficult for the non-contextual bandit to learn that the experimental condition is better on average. In contrast, the differences between conditions are large for all quartiles except Q2. Thus, the information from the student quartiles makes the problem easier for the contextual MAB algorithm, though the relatively small difference between conditions for Q2 results in the lowest overall proportion of optimal action choices. This simulation thus importantly shows a scenario that was not explored in the prior simulations, in which knowing about extra information increases the number of parameters to learn but makes learning about each of those individual parameters easier.

Even Student Distribution, Completed HW: For this scenario, there were again very high reward probabilities across all conditions, and a relatively small overall difference between conditions but larger differences between conditions for three of the four quartiles. The results from the previous simulation were mirrored here: all groups with some reward rates of less than 100% were aided by the contextual MAB algorithm.

Even Student Distribution, Completed Quickly: Finally, using the Completed Quickly outcome measure for this

second set of students, the results were still largely in favor of the contextual MAB algorithm. As the experimental condition is better on average, students in Q1 experience a large positive impact through personalization because the control condition is uniquely better for them. Q2 and Q3 also experience positive impacts, with the impact on Q2 students being larger because the difference between the two conditions is larger, which speeds learning for the contextual bandit. Conversely, Q4 students experience slightly less positive outcomes under the contextual MAB algorithm because the small difference between conditions slows learning; in comparison, the contextual MAB algorithm is more beneficial for Q4 students since the overall difference between conditions across all students is larger than the difference for Q4 students only.

Variability in real-world scenarios: Variability across trials in these scenarios showed the same trend as in the previous simulations: the non-contextual MAB algorithm typically has slightly more variation around the mean of the distribution, but only the contextual MAB algorithm shows bimodality, with some trials showing very poor performance for at least one of the groups (Figure 6).

7. DISCUSSION

Real-time adaptive algorithms can respond quickly to optimize experiences for individual students, and their expressivity for personalizing experiences increases with each additional type of student information they are given. In this paper, we have shown that this expressivity is worthwhile only when it is *necessary* for expressing the best policy to improve student outcomes. It is also especially helpful in cases where student characteristics are not uniformly distributed. In that case, an algorithm without the extra information may instead learn a policy that systematically optimizes for the majority but not for a minority group. However, when this expressivity is not necessary, it increases variability across students and also increases the time for identifying the correct policy, thus significantly decreasing the number of students assigned the best version of the technology and slightly decreasing their average outcomes. Despite this, the results based on the real-world experimental data clarify the potential benefits of personalization by demonstrating that having extra information about students can sometimes make learning easier, outweighing the negative impact of learning additional parameters.

There are several limitations to our results. First, we have focused only on discrete student features and discrete outcomes but continuous parameters are also common. For example, we might measure student scores rather than homework completion or model prior knowledge as an estimated ability parameter. If one wanted to extend these analyses to real-valued student features, one could easily incorporate them into the current modeling framework with versions of Thompson sampling for real-valued outcomes [2], and there exist metrics from a large literature for assessing whether students are treated fairly (e.g., [6]). Using real-valued parameters is unlikely to significantly impact trends in results, except that defining student groups for analyzing equitable outcomes is more difficult. Our results from our universal optimal action scenarios show that, with binary rewards, knowledge of the student features is not beneficial if it is

unnecessary for expressing the best policy. However, these results may not translate to the real-valued rewards case, where the latent student features will add to the variability in the distributions observed by the non-contextual bandit, and exploring these scenarios is an important step for future work. A second limitation is that our simulations comprise only a single student feature that influences the outcome, though in actual deployments multiple features may influence the best policy. Still, we believe that our results can guide system designers when thinking about such scenarios, especially in weighing the costs and benefits of including each possible variable.

The results from the real-world scenarios highlight the potential value of MAB algorithms for educational technologies. For almost all scenarios and groups, both types of MAB algorithms chose the optimal condition more often than if students had been assigned uniformly at random, and average rewards were in many cases very close to the optimal expected reward (i.e. if the optimal action had been chosen for all students). The absolute difference in rewards was relatively small between the two bandit types—at most 0.075—and the contextual bandit achieved at worst 12% less than the optimal expected reward for any student group. Yet the earlier simulations urge caution for incorporating student characteristics, due to (1) decreases in achieved outcomes when these characteristics are unnecessary, (2) increases in variability of performance, and (3) the systematically different treatment of students based on irrelevant characteristics. Thus, system designers should weigh the risk of not personalizing when the best policy for the minority differs from the majority with these side effects of personalization and ultimately strive to only include variables that past evidence suggests differentially impact outcomes.

One could make a number of extensions of this work for using MAB algorithms to improve and personalize educational technologies. First, contextual MAB algorithms might mitigate issues of biases when different types of students interact with an educational technology and while all are most helped by the same version of the technology, their outcomes have different distributions. For example, struggling students may complete homework later, leading the MAB algorithm’s early estimates to be non-representative of the broader population. Prior work has shown that this bias significantly worsens inference about the effectiveness of the technology as well as expected student outcomes [20]: the use of a contextual MAB algorithm could allow the system to adapt to such differences across students. Second, if the technology is used by a large number of students, the set of variables used by the contextual algorithm could be increased as more data are collected. Such a system might improve consistency across student outcomes, while still personalizing based on truly relevant features that are justified the sufficient information collected. The work in this paper both provides a starting point for considering what scenarios, algorithms, and metrics should be explored in future work, as well as guidance for system designers who would like to deploy MAB algorithms within their own technologies but are uncertain about which student characteristics, if any, to include for personalization.

8. REFERENCES

- [1] S. Agrawal and N. Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In S. Mannor, N. Srebro, and R. C. Williamson, editors, *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23, pages 39.1–39.26, Edinburgh, Scotland, 2012. PMLR.
- [2] S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on International Conference on Machine Learning*, volume 28, pages 127–135. JMLR, 2013.
- [3] V. Aleven, I. Roll, B. M. McLaren, and K. R. Koedinger. Help helps, but only so much: Research on help seeking with intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, 26(1):205–223, 2016.
- [4] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.
- [5] A. Ben-Hur and J. Weston. A user’s guide to support vector machines. In *Data mining techniques for the life sciences*, pages 223–239. Springer, 2010.
- [6] R. Binns. Fairness in machine learning: Lessons from political philosophy. In *Conference on Fairness, Accountability and Transparency*, pages 149–159, 2018.
- [7] O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2249–2257. Curran Associates, Inc., 2011.
- [8] B. Clement, D. Roy, P.-Y. Oudeyer, and M. Lopes. Multi-armed bandits for intelligent tutoring systems. *Journal of Educational Data Mining*, 7:20–48, 2015.
- [9] A. T. Corbett. Cognitive computer tutors: Solving the two-sigma problem. In M. Bauer, P. Gmytrasiewicz, and J. Vassileva, editors, *User Modeling 2001*, volume 2109 of *Lecture Notes in Computer Science*, pages 137–147. Springer, 2001.
- [10] A. Erraqabi, A. Lazaric, M. Valko, E. Brunskill, and Y.-E. Liu. Trading off rewards and errors in multi-armed bandits. In A. Singh and J. Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pages 709–717. PMLR, 2017.
- [11] J. Gardner, C. Brooks, and R. Baker. Evaluating the fairness of predictive student models through slicing analysis. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 225–234, 2019.
- [12] S. Hajian, F. Bonchi, and C. Castillo. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2125–2126, 2016.
- [13] S. Hutt, M. Gardner, A. L. Duckworth, and S. K. D’Mello. Evaluating fairness and generalizability in models predicting on-time graduation from college applications. *International Educational Data Mining Society*, 2019.
- [14] S. Jain, B. Narayanaswamy, and Y. Narahari. A multiarmed bandit incentive mechanism for crowdsourcing demand response in smart grids. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.
- [15] M. Joseph, M. Kearns, J. H. Morgenstern, and A. Roth. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*, pages 325–333, 2016.
- [16] A. S. Lan and R. G. Baraniuk. A contextual bandits framework for personalized learning action selection. In T. Barnes, M. Chi, and M. Feng, editors, *Proceedings of the Ninth International Conference on Educational Data Mining*, pages 424–429, 2016.
- [17] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 661–670. ACM, 2010.
- [18] Y.-E. Liu, T. Mandel, E. Brunskill, and Z. Popovic. Trading off scientific knowledge and user learning with multi-armed bandits. In J. Stamper, Z. Pardos, M. Mavrikis, and B. McLaren, editors, *Proceedings of the 7th International Conference on Educational Data Mining*, pages 161–168, 2014.
- [19] J. D. Lomas, J. Forlizzi, N. Poonwala, N. Patel, S. Shodhan, K. Patel, K. Koedinger, and E. Brunskill. Interface design optimization as a multi-armed bandit problem. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 4142–4153, 2016.
- [20] A. N. Rafferty, H. Ying, and J. Williams. Statistical consequences of using multi-armed bandits to conduct adaptive educational experiments. *Journal of Educational Data Mining*, 11(1):47–79, 2019.
- [21] M. Raghavan, A. Slivkins, J. V. Wortman, and Z. S. Wu. The externalities of exploration and how data diversity helps exploitation. In S. Bubeck, V. Perchet, and P. Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1724–1738. PMLR, 06–09 Jul 2018.
- [22] A. Segal, Y. B. David, J. J. Williams, K. Gal, and Y. Shalom. Combining difficulty ranking with multi-armed bandits to sequence educational content. In *Proceedings of the 19th International Conference on Artificial Intelligence in Education*, pages 317–321. Springer, 2018.
- [23] D. Selent, T. Patikorn, and N. Heffernan. ASSISTments dataset from multiple randomized controlled experiments. In *Proceedings of the Third ACM Conference on Learning at Scale*, pages 181–184. ACM, 2016.
- [24] H. Shaikh, A. Modiri, J. J. Williams, and A. N. Rafferty. Balancing Student Success and Inferring Personalized Effects in Dynamic Experiments. In *Proceedings of the 12th International Conference on Educational Data Mining*, 2019.
- [25] V. Shute. Focus on formative feedback. *Review of Educational Research*, 78(1):153–189, 2008.
- [26] S. S. Villar, J. Bowden, and J. Wason. Multi-armed bandit models for the optimal design of clinical trials:

- Benefits and challenges. *Statistical science: A review journal of the Institute of Mathematical Statistics*, 30(2):199–215, 2015.
- [27] J. J. Williams, J. Kim, A. N. Rafferty, S. Maldonado, K. Z. Gajos, W. S. Lasecki, and N. Heffernan. Axis: Generating explanations at scale with learnersourcing and machine learning. In *Proceedings of the Third ACM Conference on Learning at Scale*, pages 379–388. ACM, 2016.
- [28] J. J. Williams, A. N. Rafferty, D. Tingley, A. Ang, W. S. Lasecki, and J. Kim. Enhancing online problems through instructor-centered tools for randomized experiments. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 207:1–207:12. ACM, 2018.

What Time is It? Student Modeling Needs to Know

Ye Mao
North Carolina State Univ.
Raleigh, NC, USA
ymao4@ncsu.edu

Samiha Marwan
North Carolina State Univ.
Raleigh, NC, USA
amarwan@ncsu.edu

Thomas W. Price
North Carolina State Univ.
Raleigh, NC, USA
twprice@ncsu.edu

Tiffany Barnes
North Carolina State Univ.
Raleigh, NC, USA
tmbarnes@ncsu.edu

Min Chi
North Carolina State Univ.
Raleigh, NC, USA
mchi@ncsu.edu

ABSTRACT

Modeling student learning processes is highly complex since it is influenced by many factors such as motivation and learning habits. The high volume of features and tools provided by computer-based learning environments confounds the task of tracking student knowledge even further. Deep Learning models such as Long-Short Term Memory (LSTMs) and classic Markovian models such as Bayesian Knowledge Tracing (BKT) have been successfully applied for student modeling. However, much of this prior work is designed to handle sequences of events with *discrete timesteps*, rather than considering the continuous aspect of time. Given that time elapsed between successive elements in a student's trajectory can vary from seconds to days, we applied a Time-aware LSTM (T-LSTM) to model the dynamics of student knowledge state *in continuous time*. We investigate the effectiveness of T-LSTM on two domains with very different characteristics. One involves an open-ended programming environment where students can *self-pace* their progress and T-LSTM is compared against LSTM, Recent Temporal Pattern Mining, and the classic Logistic Regression (LR) on the early prediction of student success; the other involves a classic *tutor-driven* intelligent tutoring system where the tutor scaffolds the student learning step by step and T-LSTM is compared with LSTM, LR, and BKT on the early prediction of student learning gains. Our results show that T-LSTM significantly outperforms the other methods on the self-paced, open-ended programming environment; while on the tutor-driven ITS, it ties with LSTM and outperforms both LR and BKT. In other words, while time-irregularity exists in both datasets, T-LSTM works significantly better than other student models when the pace is driven by students. On the other hand, when such irregularity results from the tutor, T-LSTM was not superior to other models but its performance was not hurt either.

1. INTRODUCTION

Student Modeling sits at the epicenter of educational data mining. It monitors a student's progress, ability, or knowledge over a set of skills and can predict the student's future performance based on historical sequence data. In recent years, recurrent neural network architectures, such as Long Short-Term Memory (LSTMs), have become the workhorses for modeling sequence data in a variety of tasks involving sequential data, such as video processing, climate change detection, and patient disease progression prediction [20, 19, 25, 12]. Deep Knowledge Tracing [35, DKT], the first LSTM approach in student modeling, reported an impressive improvement over a classical statistical model Bayesian Knowledge Tracing [10, BKT]. Both LSTM/DKT and BKT are designed to handle sequences of events with *discrete timesteps*, not considering the *continuous* aspect of time.

On the other hand, student response time, the elapsed times between consecutive elements of a sequence can vary greatly by student, from seconds to days. Ever since the mid-1950s, student response time has been used as a preferred educational assessment to evaluate how active and accessible student knowledge is in cognitive psychology [43]. For example, it has been shown that response time reveals student proficiency [40] and there is a significant negative correlation between student average response time and student final exam score taken at the end of the semester [16]. Additionally, response time has been suggested as an indicator of student engagement in answering questions [21] as well as an important factor for predicting motivation in learning environments [9]. Also, by leveraging time information, BKT prediction performance can be improved [38, 44]. Therefore, by not taking the time intervals into consideration, the design of traditional LSTM and BKT may lead to sub-optimal performance for modeling student learning.

Previous work for modeling sequence data has explored several ways to handle time irregularity [3, 34, 8, 6] and among them, Time-aware LSTM (T-LSTM) is one of the most state-of-the-art models [3]. T-LSTM transforms time intervals between successive elements into weights and uses them to adjust the memory passed from previous moments. In this work, we apply T-LSTM to model the dynamics of student knowledge state *in continuous time* and conduct two empirical comparisons between T-LSTM and the standard LSTM, Recent Pattern Mining [23], and classical student model-

Ye Mao, Samiha Marwan, Thomas Price, Tiffany Barnes and Min Chi "What Time is It? Student Modeling Needs to Know" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 171 - 182

ing methods such as BKT and logistic regression models on two real-world data sets collected from two learning environments with very different characteristics. One is an open-ended block-based programming environment for a novice programming task where students are free to explore the environment with minimal system support or constraints. Each student's log file is a trajectory of actions with corresponding time stamps and time intervals calculated between the two consecutive student actions. The other probability tutor is *tutor-driven* in that *the tutor* decides what to do next. Each student's log file is a trajectory of student-ITS interactions. In each interaction, the tutor first *elicits* the subsequent step from a student with prompting, and when the student performs a step, the tutor records its success or failure and may give feedback (e.g. correct/incorrect markings); if the student's answer is incorrect, the tutor provides a series of hints from general to specific and the bottom-out hint tells the student exactly what to do. The interaction is ended only when a step is correctly answered and the tutor moves to the next interaction. As a result, each student's log file is a trajectory of tutor actions mixed with student's responses with corresponding time stamps. In this environment, the time intervals are calculated between the student's *first attempt* on one problem and the next. Our research question is: *By taking time-awareness into consideration, would T-LSTM outperform other traditional student modeling methods on both self-paced and tutor-driven learning environments?*

2. METHODS

2.1 Long Short-Term Memory

Long Short Term Memory [18, LSTM] is a special type of RNN which is explicitly designed to avoid the long-term dependency problem. LSTM can avoid the vanishing (and exploding) gradient problem and works tremendously well on a large variety of problems.

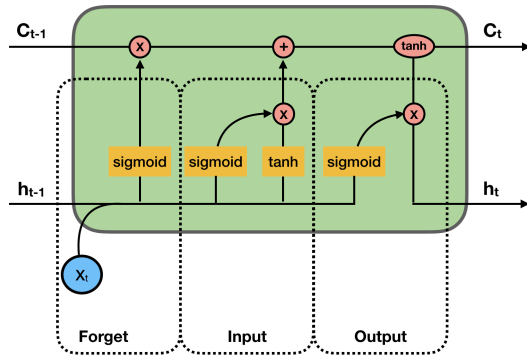


Figure 1: The Structure of a LSTM Unit

The internal structure of each LSTM module is shown in Figure 1. There are three major components: a forget gate, an input gate, and an output gate in a standard LSTM unit cell, where these components interact with each other to control how information flows. In the first step, a function of the previous hidden state h_{t-1} and the new input x_t passes through the forget gate, indicating what is probably irrelevant and can be taken out of the cell state. The for-

get component will calculate a weight f_t between 0 to 1 for each element in hidden state vector C_{t-1} . An element with a weight of 0 should be completely forgotten whereas an element with a weight of 1 needs to be entirely remembered. The formula to calculate f_t is shown below where W_f and b_f are the weights and intercepts, respectively, for the forget component.

$$f_t = \text{sigmoid}(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

There are two steps involved in input component's calculation. In the first step, a *tanh* layer calculates a candidate vector \tilde{C}_t that could be added to the current hidden state. In the second step, the input components calculate a weight vector i_t (ranging from 0 to 1) to determine to what extent \tilde{C}_t should update the current memory state.

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2)$$

$$i_t = \text{sigmoid}(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

With the forget and input components, the module is able to throw away the expired information in the previous cell state by calculating $C_{t-1} \cdot f_t$, and process new information by computing $\tilde{C}_t \cdot i_t$. Consequently, the formula to update the current memory cell is shown below. Note that the current memory cell state C_t is then passed to the next LSTM module.

$$C_t = C_{t-1} \cdot f_t + \tilde{C}_t \cdot i_t \quad (4)$$

Finally, the output component is simply an activation function that filters elements in C_t . The C_t can be converted to a value between -1 to 1 by the tanh function. The output component calculates a weight vector

$$o_t = \text{sigmoid}(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

that determines how much information is allowed to be revealed.

$$C_t = o_t * \tanh(C_t) \quad (6)$$

With such a gated structure, LSTM is capable of handling long-term dependencies.

2.2 Time-Aware Long Short Term Memory

The standard LSTM assumes that the elapsed times between elements of a sequence are uniformly distributed, and therefore it is designed to handle sequences with *discrete timesteps*. However, in the educational domain, the interval between two consecutive steps during a student trajectory can span from seconds to days. In general, the events that occurred long ago tend to have less impact to the current state and thus we should properly reduce their contributions. Therefore, it is important to consider the elapsed time when predicting the current event's output. In this work, we applied Time-aware LSTM [3, T-LSTM], which is proposed to handle the temporal dynamics of sequential data with time irregularities, to model student knowledge states *in continuous time*.

The T-LSTM architecture is shown in Figure 2. To fit in our domain, we represent the input sequence by the student trajectories. Apart from the three gates in standard LSTM: forget, input, and output; T-LSTM also integrates the time elapsed between successive records into the network architecture, and we call this as the time decay component. The information stored in the memory of the previous

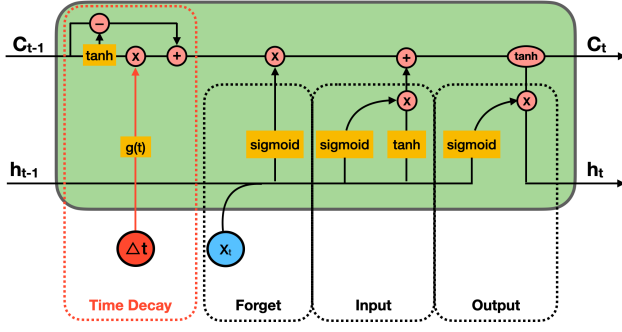


Figure 2: The Structure of a T-LSTM Unit

hidden state C_{t-1} is decomposed into two parts: long-term memory and short-term memory. Without losing the long-term memory contained in C_{t-1} , the time decay component mainly plays a role to adjust the short-term memory by employing the elapsed time between successive steps. If the gap between two steps is significantly huge, e.g. few hours in our domain, it means there has been a long time with no interaction between students and the tutor/computer. In that case, there is not much point to heavily rely on the previous short-term memory to predict the current output. In the framework of T-LSTM, a non-increasing function of the elapsed time is applied to transform the time duration into an appropriate weight. And in this work, we applied $g(\Delta t) = 1/\log(e + \Delta t)$ to get the corresponding weights.

The following calculations are involved in the time decay component of T-LSTM. First, short-term memory C_{t-1}^S is calculated.

$$C_{t-1}^S = \tanh(W_d \cdot C_{t-1} + b_d) \quad (7)$$

The long-term memory can be obtained by deducting short-term memory from the previous hidden state.

$$C_{t-1}^L = C_{t-1} - C_{t-1}^S \quad (8)$$

Then C_{t-1}^S is discounted by the elapsed time weight to obtain the discounted short-term memory \hat{C}_{t-1}^S .

$$\hat{C}_{t-1}^S = C_{t-1}^S * g(\Delta t) \quad (9)$$

Finally, the adjusted previous hidden state C_{t-1}^* is composed by adding long-term memory and discounted short-term memory.

$$C_{t-1}^* = C_{t-1}^L + \hat{C}_{t-1}^S \quad (10)$$

The following parts are very similar to standard LSTM. Following the steps in Section 2.1, we first calculate the forget gate f_t , candidate vector \tilde{C}_t and input gate i_t by applying Equation (1), (2) and (3). For the calculation of the current memory cell state C_t , the adjusted previous hidden state C_{t-1}^* instead of C_{t-1} is applied in the T-LSTM framework.

$$C_t = C_{t-1}^* \cdot f_t + \tilde{C}_t \cdot i_t \quad (11)$$

The final output for the current state can be achieved using the following Equation (6). In this work, we investigate the effectiveness of T-LSTM via the early prediction of both student success and learning gains. As far as we know, no prior studies have explored T-LSTM on both computer-based programming systems and intelligent tutoring systems.

2.3 Recent Temporal Pattern Mining

The Recent Temporal Pattern mining (RTP) framework [2] was originally proposed to find predictive patterns from complex multivariate time series data. This framework first converts time series into time-interval sequences of temporal abstractions, and then constructs more complex temporal patterns backwards. The following part will explain how the RTP framework is applied in our work.

Multivariate State Sequences: We denote a **State** S as (F, V) , where F is a temporal feature and V is the value for feature F at a given time and the **State Interval** E is denoted as (F, V, s, e) , where s and e refer to the *start* and *end* times of the state (F, V) . Thus, we can convert each student's data x_i into a corresponding Multivariate State Sequence (MSS) z_i by sorting all the state intervals by their start times: $z_i = \langle E_1, E_2, \dots, E_n \rangle : E_j.s \leq E_{j+1}.s, j \in \{1, \dots, n-1\}$. And we apply two temporal relations in this work: 1) E_i **before(b)** E_j : When E_i ends before the start of E_j ($E_i.e < E_j.s$); 2) E_i **co-occurs(c)** with E_j : When E_i and E_j have some overlap ($E_i.s \leq E_j.s \leq E_i.e$).

Recent Temporal Patterns: Here, we call a state interval $E = (F, V, s, e)$ a *Recent State Interval* of MSS z_i if: 1) E is the last state interval for feature F ; that is, for all $E' = (F, V', s', e')$, we have $E'.e \leq E.e$; or 2) E is less than g time units away from the end time of the last state interval: $z_i.end$; that is, $z_i.end - E.e \leq g$.

Given an MSS z_i , a temporal pattern $P = (\langle S_1, \dots, S_n \rangle, R)$, and a maximum gap parameter g , we say P is a recent temporal pattern (RTP) in z_i , denoted $R_g(P, z_i)$, if all 3 of the following conditions hold: 1) z_i contains P , where $P \in z_i$ if: (a) z_i contains all k states of P , and (b) all temporal relations of P are satisfied in z_i ; 2) $S_n = (F_n, V_n)$ matches a recent state interval in z_i ; and 3) Every consecutive pair of states in P maps to a state interval less than g time units apart. That is, each pair of temporal sequences should not be g time units apart. In short, parameter g forces patterns to be close to the end of the sequence z_i , and forces consecutive states to be close to each other.

Mining Algorithm: Taking student success classification as an example, we will have two sets of labeled MSSs: $Z_1 = \{z_i : y_i = 1\}$ for all *unsuccessful* sequences and $Z_0 = \{z_j : y_j = 0\}$ for all *successful* ones. Given Z_1 , the mining algorithm applies a level-wise search to find frequent RTPs. More specifically, it first starts with all frequent 1-RTPs, and then extends the patterns by adding a new state to each sequence, one at a time, until no new patterns are discovered. That is, at each level k , the algorithm finds frequent $(k+1)$ -RTPs by repeatedly extending k -RTPs through Backward candidate generation, and the Counting phase, as described below.

Backward $(k+1)$ -pattern candidates are generated from a k -pattern $P = (\langle S_1, \dots, S_k \rangle, R)$, by adding a new frequent state, S_{new} , to the beginning of the sequence to create $P' = (\langle S_{new}, S_1, \dots, S_k \rangle, R')$. Then we specify the new before (b) or co-occurs (c) relations R' between S_{new} and all original k states, restricted by the following two criteria: 1) Two state intervals of the same temporal feature cannot co-occur.

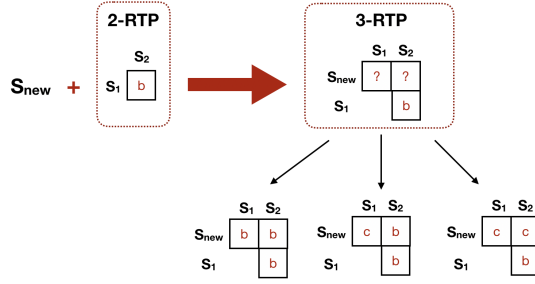


Figure 3: An example of generating 3-patterns out of a single 2-RTP, by appending a new state.

That is, if $S_{new} \cdot F = S_i \cdot F$ for $i \in \{1, \dots, k\}$, then $R'_{new,i} \neq c$. 2) Since the state sequence in pattern P is sorted by the start time of the states, once a relation becomes *before*: $R'_{new,i} = b$ for any $i \in \{1, \dots, k\}$, all of the following relations have to be *before*, so $R'_{new,j} = b$ for $j \in \{i + 1, \dots, k\}$.

In the Counting phase, candidate $(k + 1)$ -patterns are removed if they do not meet the minimum support threshold by occurring at least σ times as RTPs in Z_1 . The same procedure is carried out for Z_0 . Finally, we combine all the frequent RTPs into a final Ω set of RTPs.

Binary Matrix Transformation: We transform each MSS $z_i \in Z$ into a binary vector v_i of size $|\Omega|$, such that each 0 and 1 indicates whether the pattern $P_j \in \Omega$ is a recent temporal pattern in Z_i or not. This will result in a binary matrix of size $N \times |\Omega|$, which represents our original dataset.

2.4 Bayesian Knowledge Tracing

BKT is a student modeling method extensively used in ITSs. Figure 4 shows a graphical representation of the model and a possible sequence of student observations. The shaded nodes S represent hidden knowledge states. The unshaded nodes O represent observation of students' behaviors. The edges between the nodes represent their conditional dependence.

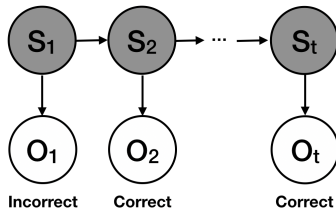


Figure 4: The Bayesian network topology of the standard Knowledge Tracing model

Fundamentally, the BKT model is a two-state Hidden Markov Model [11, HMM] characterized by five basic elements: 1) N , the number of different types of hidden state; 2) M , the number of different types of observation; 3) Π , the initial state distribution $P(S_0)$; 4) T , the state transition probability $P(S_{t+1}|S_t)$ and 5) E , the emission probability $P(O_t|S_t)$. Note that both N and M are predefined before training occurs, while Π , T and E are learned from the students' observation sequence.

Conventional BKT assumes there are two types of hidden knowledge states ($N=2$) corresponding to student knowledge states of *unlearned* and *learned*. It also assumes there are two types of student observation ($M=2$) corresponding to student performance of *incorrect* and *correct*. BKT makes two assumptions about its conditional dependence as reflected in the edges in Figure 4. The first assumption BKT makes is a student's knowledge state at a time t is only contingent on her knowledge state at time $t - 1$. The second assumption is a student's performance at time t is only dependent on her current knowledge state. These two assumptions are captured by the state transition probability T and the emission probability E . In the context of student learning, BKT further defines five parameters:

Prior Knowledge = $P(S_0 = \text{learned})$
Learning Rate = $P(\text{learned} | \text{unlearned})$
Forget = $P(\text{unlearned} | \text{learned})$
Guess = $P(\text{correct} | \text{unlearned})$
Slip = $P(\text{incorrect} | \text{learned})$

In order to apply BKT to our dataset, we captured and mapped all students' actions based on the learning opportunities of knowledge components (KCs) step by step. For each of the KC, the Baum-Welch algorithm (or EM method) is used to iteratively update the model's parameters until a maximized probability of observing the training sequence is achieved.

3. EXPERIMENTS

In this work, we explored different student modeling tasks based on characteristics of two different learning environments. One was the task of early prediction of student success in an open-ended, self-paced programming environment while the other is the task of early prediction of student learning gains within a tutor-paced probability tutor.

3.1 Predicting Student Success on iSnap

3.1.1 iSnap

iSnap¹ is an extension to Snap! [15], a block-based programming environment, used in an introductory computing course for non-majors in a public university in the United States [37]. iSnap extends Snap! by providing students with data-driven hints derived from historical correct student solutions [36]. In addition, iSnap logs all students actions while programming (e.g. adding or deleting a block), as a *trace*, allowing us to detect the sequences of all student steps, as well as the *time* taken for each step. In this work, we focused on one homework exercise named Squirrel, derived from the BJC curriculum [15]. In Squirrel, students are asked to write a procedure that draws a square-like spiral. As shown in Figure 5, correct solutions require procedures, loops, and variables using at least 7 lines of code. We collected students' data for Squirrel from Spring 2016, Fall 2016, Spring 2017, and Fall 2017. We excluded students who requested hints from iSnap to eliminate factors that might affect students' problem-solving progress, leaving a total of 65, 38, 29,

¹All tutors and assignments names have been blinded for anonymous review

and 39 student code traces from each semester, respectively. The detailed statistics for iSnap dataset are shown in Table 1.

The data collected from iSnap consists of a code trace for each student's attempt. This code trace represents a sequence of timestamped snapshots of student code. We used an expert feature detector (EFD), described in [49], that automatically detects 7 features of a correct solution in a student snapshot. For example, for each snapshot in a student code trace, the EFD outputs a *feature state*, which is a series of 0s and 1s (e.g. 10000001) indicating the absence or presence of each feature, such that *feature-state: 10000001* shows that feature 1 and feature 7 are present, while the other 5 features are not. We ran the expert-feature detector to tag each snapshot in all 171 code traces, making a total of 31,064 tagged snapshots.

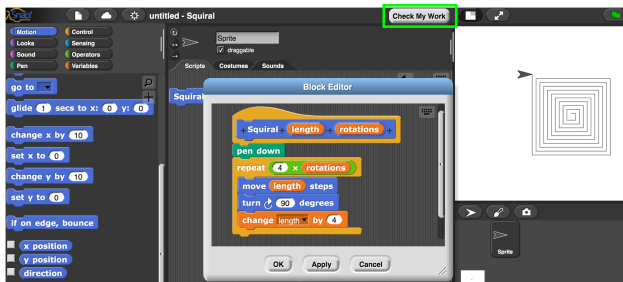


Figure 5: The iSnap interface, with the blocks palette on the left, the output stage on the right, the scripting area in the middle, and the hints button on top.

3.1.2 Student Success

In the context of iSnap, all the models were measured on the task of predicting student success. We classify the students who finished the programming assignment in one hour or less and got full credit as *successful* and labeled with “0”, those who either failed to complete or submit the assignment within one hour as *unsuccessful*, labeled with “1”. The one-hour cutoff was chosen based on a distribution showing that the vast majority of students (around 94%) who complete the assignment with full credit do so within one hour. Thus, each *trajectory* is assigned one ground truth label based on whether the student finished the assignment successfully or unsuccessfully. As a result, we refer to this task as the early prediction task for student success. Based on this definition, 59 of 171 students are in the *successful* group, and the remaining 112 are in the *unsuccessful* group. Note that this is a homework assignment that counts for only a small portion of a student's overall grade, and this behavior (of not attempting to obtain full credit) is typical in this introductory level.

To predict student success, we are given the first *up to n minutes* of a student's sequence data and our goal is to predict whether the student will successfully complete the programming assignment at any given point in the remainder of the sequence. To conduct this task, we left-aligned all the students' trajectories by their starting times and our *observation window* (the part of data used to train and test dif-

ferent machine learning models) includes the sequences from the very beginning to the first *n* minutes. If a student's trajectory is less than *n* minutes, our observation window will include their entire sequence *except* the last one.

3.1.3 Four Models

In the task of early prediction of student success, we have four models involved: Logistic Regression (LR), RTP, LSTM and T-LSTM. Note that BKT is not included here because for the open-ended domain like iSnap, there are no predefined steps or knowledge components that students must achieve to complete a given program. Thus, it is hard to map student actions on iSnap to learning opportunities defined in BKT.

Logistic Regression (LR): Since LR do not handle sequence data directly, we used a “Last Value” approach to treat the last measurement of each attribute within the given observation window as the input to train models. For early prediction settings, we truncated all the sequences in the training dataset in the same fashion as the testing dataset and then applied the Last Value approach on the truncated training dataset. For example, when our observation window is 6 minute, we apply the last value before 6 minutes for each sequence and treat them as inputs for LR.

RTP: For the RTP-based model, we first used RTP mining to generate the binary matrix and then applied LR to learn from the generated binary matrix. For early prediction, we only apply the *truncated* training sequences included in observation window to find RTPs. For example, for our 6-minute observation window, only the first 6 minutes of sequences were used for pattern extraction.

LSTM and T-LSTM: For LSTM the input is a multivariate temporal sequence from student work, and the output from the last step is used to make a prediction. While for T-LSTM, we also feed it with another sequence indicating time intervals for each student. As shown in Table 1, the time intervals of iSnap range from 1 to 291 seconds across four semesters, with $\mu = 0.613$ and $\sigma = 0.217$ for the overall decayed intervals. For both LSTM and T-LSTM, we used one hidden layer with 128 hidden neurons and set the maximum length to accommodate the longest sequence in our data. Typically for deep learning models, the whole multivariate time series from student sequence data is used as input data. However, for early prediction, only those events happening within our observation window from each sequence were used.

3.2 Predicting Learning Gains on Pyrenees

3.2.1 Pyrenees

Pyrenees is a web-based ITS teaching probability, which covers 10 major knowledge components (KCs), such as the Addition Theorem, the Complement Theorem, and Bayes' Rule, etc. Domain experts both identified the 10 KCs and labeled each step/exercise with the corresponding KCs, kappa > 0.9. Figure 6 shows the interface of Pyrenees which consists of a problem statement window, a variable window, an equation window, and a tutor-student dialogue window. Through the dialogue window, Pyrenees provides messages to the students. It can explain a worked example or prompt

Table 1: Detailed data statistics for iSnap, including total steps, total time spent in minutes, time intervals in seconds, corresponding decayed time intervals, and the success labels distribution for each of the four semesters.

Semester	Total Steps				Total Time (minutes)				Time Intervals (seconds)				Decayed Time Intervals		Success Labels	
	min	max	median	mean(std)	min	max	median	mean(std)	min	max	median	mean (std)	mean(std)		S	U
S16	10	1024	169	199 (175)	0.533	95.667	20.733	22.777 (17.149)	1	209	2	6.739 (13.75)	0.628 (0.217)		23	42
F16	28	884	121	167 (168)	3.283	119.083	16.325	22.379 (24.177)	1	189	3	7.919 (14.12)	0.594 (0.217)		15	23
S17	15	439	75	112 (94)	2.817	62.983	14.167	16.347 (11.872)	1	177	3	8.512 (16.14)	0.599 (0.225)		12	17
F17	10	2276	100	219 (376)	1.65	189.667	19.1	28.224 (33.869)	1	291	3	7.597 (15.61)	0.609 (0.215)		9	30

the student to complete the next step. Students can enter their inputs in the text area. Any variable or equation that is defined through this process is displayed on the left side of the screen for reference. Pyrenees can also provide on-demand hints. The bottom-out hint tells the student exactly how to solve a problem. Different from iSnap, the Pyrenees tutor provides immediate feedback for correct/incorrectness whenever an answer is submitted.

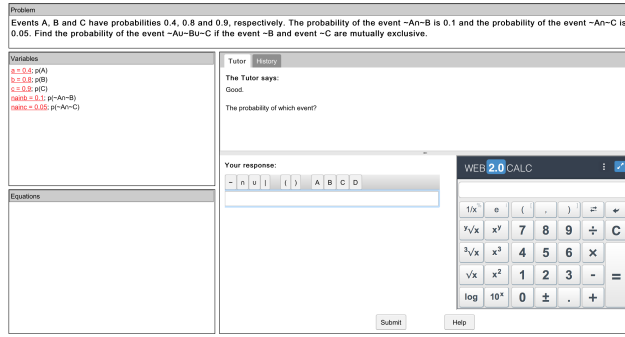


Figure 6: The Pyrenees interface, with the problem statement on the top, the variable window in the middle, the equation window at the bottom, and the dialog window on the right.

When training on Pyrenees, students were required to complete 4 phases: 1) pre-training, 2) pretest, 3) training, and 4) post-test. During the pre-training phase, all students studied the domain principles from a probability textbook. The students then took a pretest which contained 10 problems. The textbook was not available. Students were not given feedback on their answers, nor were they allowed to go back to earlier questions. During the training phase, students received the same 12 training problems in the same order on Pyrenees. Each domain concept was applied at least twice. The minimum number of steps needed to solve each training problem ranged from 10 to 50. The number of domain principles required to solve each problem ranged from 3 to 10. Finally, all of the students took a post-test with 20 problems. Both pretests and post-tests were graded in a double-blind manner by a single experienced grader (not the authors), and were normalized in the range of [0,1]. We collected six semesters of data from Pyrenees, including Fall 2016, Spring 2017, Fall 2017, Spring 2018, Fall 2018, and Spring 2019. The overall dataset comprises 102,948 data points from 1190 students, with 207, 159, 215, 161, 261 and 187 from each semester, respectively. The detailed statistics for Pyrenees dataset are shown in Table 2.

3.2.2 Quantized Learning Gain

In the context of Pyrenees, we applied all the models for student learning gains prediction. The concept of *learning gain* is formally defined as the difference between the skills, competencies, content knowledge and personal development demonstrated by students at two points in time [28]. we used a qualitative measurement called *Quantized Learning Gain* [24, QLG] to determine whether a student has benefited from our learning environment. QLG is a *binary* qualitative measurement on students' learning gains from pretest to the posttest: high vs. low. To infer QLGs, students were split into "low", "medium", and "high" based on whether they scored below the 33rd percentile, between the 33rd and 66th percentile, or higher than the 66th percentile in pre-test and post-test respectively. Once a student's pre- and post-test performance groups are decided, the student is a "high" QLG if he/she moved from a lower performance group to a higher performance group from pre-test to post-test or remained in "high" performance groups; whereas a "low" QLG is assigned to the student if he/she either moved from a higher performance group to a lower performance group from pre-test to post-test, or stayed at a "low" or "medium" groups (as shown in Figure 7). In Figure 7, solid lines represented the formation of the *high* QLG groups and dashed lines represents the formation of the *low* QLG groups, and they will be coded with "1" and "0" respectively for QLG prediction. As a result, we have 487 of 1190 students in the *high* learning gain group, and the remaining 703 students in the *low* learning gain group.

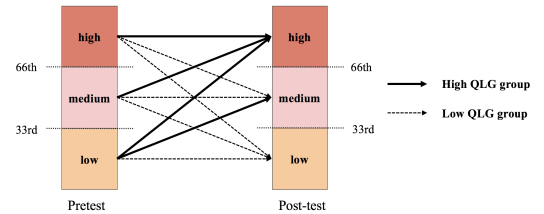


Figure 7: Quantized Learning Gain

Students usually need to spend 2-4 hours to complete the Pyrenees tutor. Thus we are given the first *up to n percent* of a student's sequence data to predict student QLG, and our goal is to predict whether the student will benefit from our tutoring system in the end. As with the success prediction in iSnap, we left-aligned all the students' trajectories by their starting times and our *observation window* includes the data from the very beginning to the first *n* percent of the whole sequence.

Table 2: Detailed data statistics for Pyrenees, including total steps, total time spent in hours, time intervals in seconds, corresponding decayed time intervals, and the QLG labels distribution for each of the six semesters.

Semester	Total Steps				Total Time (hours)				Time Intervals (seconds)				Decayed Time Intervals mean (std)	QLG Labels	
	min	max	median	mean(std)	min	max	median	mean (std)	min	max	median	mean (std)		low	high
F16	12	144	78	75 (25)	0.545	173.553	4.142	15.039 (25.56)	1	542136	31	731.799 (10876.82)	0.298 (0.11)	80	127
S17	59	152	88	94 (23)	0.642	240.661	2.643	17.492 (38.29)	1	861636	25	685.094 (14329.39)	0.314 (0.11)	59	100
F17	38	148	113	105 (25)	0.773	576.055	5.100	24.335 (64.29)	1	1287547	24	844.083 (20941.73)	0.313 (0.11)	105	110
S18	23	138	73	71(21)	0.587	135.597	2.682	9.431(18.33)	1	354272	27	486.703 (7021.54)	0.307 (0.10)	47	114
F18	26	162	86	88 (23)	0.679	165.559	4.024	14.914 (22.54)	1	438986	28	613.861 (8924.83)	0.301 (0.10)	98	163
S19	12	138	81	83 (21)	0.571	170.116	4.613	16.909 (27.56)	1	609641	28	738.505 (11439.02)	0.305 (0.11)	98	89

3.2.3 Four Models

In the task of early prediction of student QLG, we have four models involved: LR, BKT, LSTM and T-LSTM. Note that we do not compare RTP here because, in Pyrenees, students’ responses are determined not only by their underlying knowledge state, but also by the pre-designed turn-taking nature of the system, which could obscure the temporal patterns found by RTP.

Logistic Regression (LR): As with student success prediction, the “Last Value” approach was applied to the non-temporal LR for the task of predicting student learning gains, as well as the early prediction setting. For example, when the training data is the *first 30% sequence*, only the last value before 30% of each sequence was applied for both training and testing.

BKT: To train the BKT model for QLG prediction, two steps were involved. In the first step, the probability of a student being in the *learned* state on each KC at the last attempt was learned from the BKT model. And in the second step, the output of the first step was computed as features for our prediction tasks. That is, the number of features involved here equals to the total number of KCs involved. The logistic regression was applied to predict QLG. As with early prediction setting of student success, only the *truncated* training sequences were applied to learn student learning probabilities from BKT.

LSTM and T-LSTM: In order to better compare LSTM and T-LSTM performance with BKT, the same two types of features were applied here for QLG prediction: 1) the assignment of KCs corresponding to each step, and 2) student performance at each step, i.e, correct or incorrect. As shown in Table 2, the time intervals of Pyrenees range from 1 second to 14 days across the six semesters, with $\mu = 0.307$ and $\sigma = 0.107$ for overall decayed intervals. For both LSTM and T-LSTM, we used one hidden layer with 64 hidden neurons and also set the maximum length to accommodate the longest sequence in our data. Again, only those events happening within our observation window from each sequence were applied for training and testing of early prediction.

3.3 Evaluation Metrics

Our models in this work were evaluated using Accuracy, Precision, Recall, F1 Score, and AUC (Area Under ROC curve). Accuracy represents the proportion of students whose labels were correctly identified. Precision is the proportion of students who were predicted to be successful by each model who were actually in the *successful* (or *high* QLG) group. Recall tells us what proportion of students, who will actu-

ally be unsuccessful (or in *low* QLG group), who were correctly recognized by the model. F1 Score is the harmonic mean of Precision and Recall that sets their trade-off. AUC measures the ability of models to discriminate groups with different labels. Given the nature of the tasks, we mainly use Accuracy and AUC to compare different models. Finally, it is important to emphasize that all models were evaluated using semester-based temporal cross-validation for both tasks, which just applied data from previous semesters for training and is a much stricter approach for time series data than the standard cross-validation.

4. RESULTS

4.1 Predicting Student Success in iSnap

Table 3 shows the performance of all models using the first-6-minute training sequences to predict students’ success in the programming task. The first column indicates the models including majority baseline model using simple Majority vote, Logistic Regression (LR), RTP, LSTM and T-LSTM. Columns 2-5 report all of the models’ performance for the first-6-minute observation window. We evaluated the models on different metrics including Accuracy, Precision, Recall, F1 and AUC score; note that we ignored the Precision, Recall and F1-measure of the simple Majority baseline. The last column reports the mean AUC score of all models from 0 - 20 minutes, with standard deviations between brackets. At first-6-minute, we can observe that T-LSTM outperforms all the other models and it contributes the highest score on every measurement except that the best Recall comes from RTP. LSTM and RTP have very similar performance at first-6-minute, and both of them get better performance than LR except on Precision and AUC. On the other hand, when comparing the overall AUC score among all the models, T-LSTM still achieves the highest score. These results suggests T-LSTM can better learn the difference between successful/unsuccesful groups with the help of time-awareness.

Figures 8 (a) and (b) report Accuracy and AUC performance respectively for all models predicting student success. For each graph, we vary the observation window from the first 2 minutes up to 20 minutes. As shown in Table 1, students generally take 10 to 60 minutes to complete the task and thus we took a measurement every 2 minutes for the first 10 minutes to generate the early stage predictions for each model. T-LSTM is in red, LSTM in blue, RTP in purple, LR in green, and majority baseline in black. Both Figures 8 (a) and (b) show that T-LSTM was the best model for student programming success prediction as it stays on the top across all sizes of the observation window. It is not surpris-

Table 3: iSnap Student Success Prediction at First-6-minute and Overall Time (0 - 20 minutes)

Models	first-6-minute					Overall
	Accuracy	Precision	Recall	F1-measure	AUC	AUC
Majority	0.6604	-	-	-	0.5000	0.5000
LR	0.6038	0.8333	0.5000	0.6250	0.6528	0.7123(± 0.08)
RTP	0.6792	0.7195	0.8429	0.7763	0.6020	0.6948(± 0.09)
LSTM	0.6792	0.7368	0.8000	0.7671	0.6222	0.6755(± 0.09)
T-LSTM	0.7358	0.875	0.7000	0.7778	0.7528	0.7512(± 0.07)

Note: best model on each metric in **bold**

ing that generally for all the models (except majority baseline), the longer the observation windows, the better performance. This is because the training data includes more and more information and students get closer to their final state. The fact that the best prediction comes from T-LSTM really suggests that during the self-paced programming task, taking time-awareness into consideration brings us closer to the truth of the student learning process, especially for the early stage (first 10 minutes). However, this is only one observation from one programming task and more research is needed to understand the full nature of the benefits of time-awareness.

4.2 Predicting Learning Gains in Pyrenees

Table 4 shows the performance of all models using the first-30%-sequence to predict students' QLG on the probability tutor. The first column indicates the models including majority baseline model using simple Majority vote, LR, BKT, LSTM, and T-LSTM. Columns 2-5 report the all of the models' performance at the first-30%-sequence observation window. As with Table 3, we evaluated the models on Accuracy, Precision, Recall, F1 and AUC score and ignored the Precision, Recall and F1-measure for the simple Majority baseline. The last column reports the mean AUC score of all models from 0 - 100% sequence, with standard deviations between brackets. When only applying the first-30%-sequence, T-LSTM generates the best performance on every measurement except Recall and F1, where the best Recall is from LR and best F1 from LSTM. Comparing the two deep learning models with classic BKT, we can observe that both LSTM and T-LSTM outperform BKT across all metrics. For the overall AUC performance, LSTM and T-LSTM have very similar scores and are equally good. And still, they achieve higher mean AUC scores than BKT, with a lower standard deviation. Despite the similar overall performance from the two deep learning models, the better early prediction of T-LSTM suggests that time-awareness can help to understand student learning states earlier.

The early prediction results for student learning gains in probability are reported in Figure 9. BKT is in purple, and as in Figure 8, T-LSTM, LSTM, and LR are in red, blue and green, respectively. For each graph, the results are measured at every 10% increment of the sequence length. Generally speaking, the three models (BKT, LSTM and T-LSTM) generate better results as the sequence length increases. Both Figures 9 (a) and (b) show that the two deep learning models outperform BKT for probability, no matter on Accuracy or AUC score. While between LSTM and T-LSTM, there is not a clear winner. Sometimes T-LSTM gets better per-

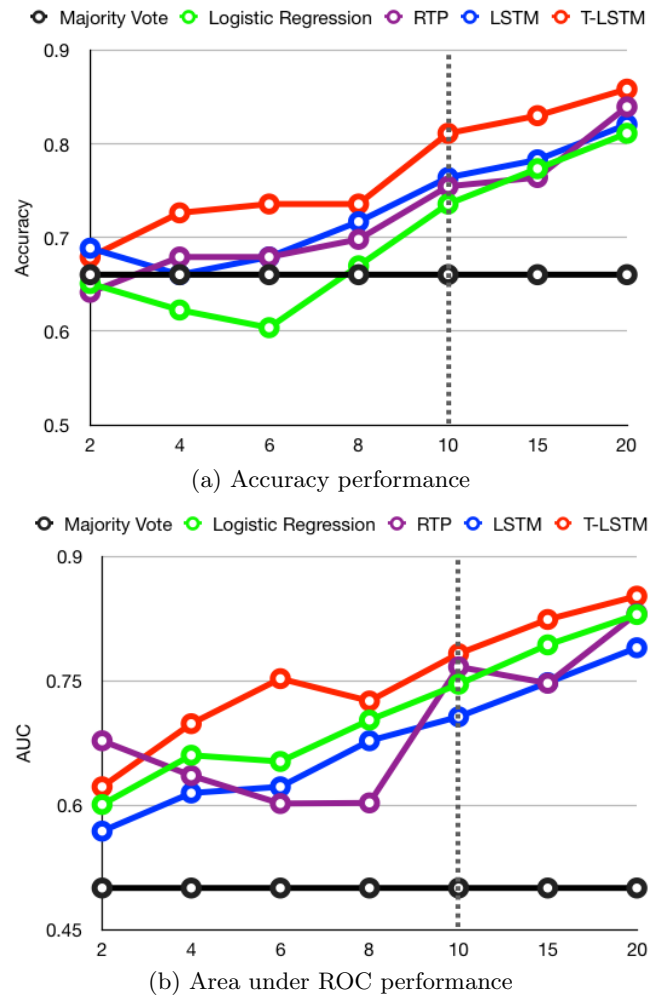


Figure 8: Student Success Early Prediction on iSnap

Table 4: Pyrenees Student QLG Prediction at First-30%-minutes and Overall Time (0 - 100%)

Models	first-30%-sequence					Overall AUC
	Accuracy	Precision	Recall	F1-measure	AUC	
Majority	0.5860	-	-	-	0.5000	0.5000
LR	0.5839	0.5893	0.9566	0.7293	0.5066	0.4957(± 0.01)
BKT	0.6022	0.6113	0.8819	0.7221	0.5442	0.5690 (± 0.03)
LSTM	0.6226	0.6188	0.9271	0.7422	0.5594	0.6013 (± 0.02)
T-LSTM	0.6328	0.6322	0.8924	0.7401	0.5789	0.5950 (± 0.02)

Note: best model on each metric in **bold**

formance on Accuracy (from 10% to 30%) while sometimes LSTM slightly outperforms T-LSTM (from 40% to 70%). Overall, LSTM and T-LSTM generate very similar results on predicting student QLG; and T-LSTM generally has better performance on the very early stage.

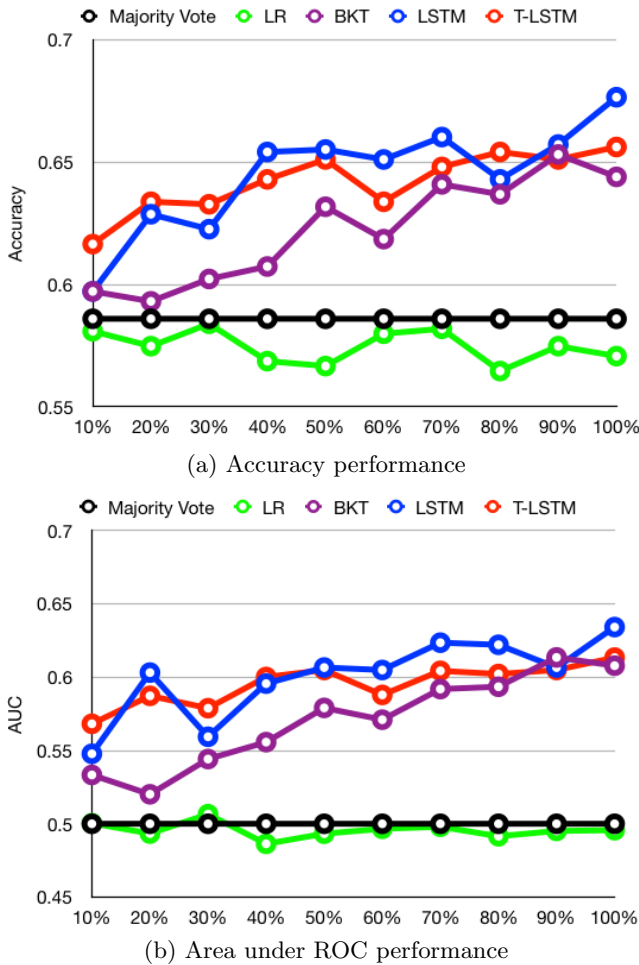


Figure 9: Student Learning Gain Early Prediction on *Pyrenees*

5. RELATED WORK

Student modeling has been widely and extensively explored in previous research. For example, prior research has proposed a series of approaches based on logistic regression including Item Response Theory (IRT) [42], Learning Factor Analysis [5], Learning Decomposition [4], Instructional Factors Analysis [7], Performance Factors Analysis [33], and Recent-Performance Factors Analysis [14]. These models were implemented with different parameters to better understand and model student learning and were shown to be very successful.

BKT [10] is one of the most widely investigated student modeling approaches. It models a student's performance in solving problems related to a given concept using a binary variable (i.e., correct, incorrect) and continually updates its estimation of the student's learning state for that concept. Many extensions of BKT have been proposed to capture the complex and diverse aspects of student learning. Pardos and Heffernan [31] explored individualized prior knowledge parameters based on students' overall competence. Their results showed that the proposed model outperformed conventional BKT in predicting students' responses to the last question at the end of the entire training. They later introduced problem difficulty to BKT and found substantial performance improvement in predicting student step-by-step responses over BKT [32]. Additionally, Yudelson et al. [48] parameterized student learning rates in BKT models and the results showed that the new model outperformed conventional BKT in predicting whether the students' next responses were going to be correct/incorrect. Baker et al. [1] investigated contextualized guess and slip rates to deal with the issues of identifiability and model degeneracy commonly observed in conventional BKT. Their results suggested that the proposed models achieved better performance in predicting students' next-step response than BKT. However, in this study, BKT-based models cannot be directly applied to our open-ended programming tasks, because of the adversity of mapping students' time-various actions step by step.

In recent years, extensive research has been conducted on deep learning models, especially Recurrent Neural Networks (RNN) or RNN-based models such as LSTM. These deep recurrent models have shown great success in many domains

such as speech recognition [17], language translation [26], video classification [29], and rainfall intensity prediction [46], etc. Their success in all these domains has opened up a new line of research in educational data mining [35, 41, 22, 45, 47, 24, 30]. Mao et al. [27] have shown that LSTM has superior performance on the early prediction of student learning gains compared with classic BKT-based models. For the task of predicting students' responses to exercises, LSTM was shown to outperform conventional BKT [35] and Performance Factors Analysis [33]. However, RNN and LSTM did not always have better performance when the simple, conventional models incorporated other parameters. For example, Khajah et al. [22] investigated what statistical regularities neural networks can exploit that BKT cannot, and showed that BKT with relaxed assumptions can outperform LSTM. Wilson et al. [45] also show that Bayesian extensions of simple IRT-based models are also equal to or outperform RNN-based models on a variety of datasets.

While most of the previous studies on student modeling focus on predicting students' success and failure in the next-step attempt, some research has used student-tutor interaction data to predict student post-test scores [13, 39]. In this work, we explored the early prediction of student success and learning gains for a computer-based programming system and an intelligent tutoring system, respectively.

6. CONCLUSIONS

Early prediction of student learning state is a crucial component of student modeling, since it allows tutoring systems to intervene by providing needed support, such as a hint, or by alerting an instructor. Both prediction tasks involved in this work are challenging because: 1) the open-ended nature of iSnap hinders the prediction of student final success, and 2) it is extremely hard to track whether a student benefits from a tutoring system or not even in a well-defined domain like Pyrenees. In this work, we investigated the effectiveness of a time-aware model, T-LSTM on the two different prediction tasks and compared it with other student modeling methods including LSTM, RTP, logistic regression models, and BKT. Our results show that T-LSTM consistently outperforms the other models such as LSTM, RTP, and non-temporal logistic regression on the task of predicting student success in iSnap, at all observation windows from first 2 minutes to 20 minutes. On the other hand, for the task of predicting student learning gains in Pyrenees, T-LSTM does not outperform the other models. More specifically, T-LSTM outperforms LSTM and BKT on the early stage with only 30% of the student sequences, and afterward time-awareness does not help much when more data is available. One possible explanation behind this is that in a well-defined domain, the whole learning process is mainly driven by the tutor, which makes the elapsed time less important to student learning gains especially when the step-level performance is available. However, in the open-ended programming environment, students are self-prompted to complete an assignment; and therefore the amount of time they stayed in a state really matters to understand their learning. And therefore, T-LSTM can generate better performance by modeling the student dynamics of knowledge *in continuous time* than other methods in *discrete timesteps*.

One limitation of this work is that we only explored one im-

portant student modeling task in each learning environment. An important direction for future work is to investigate the time-aware model on other student modeling tasks in both learning environments to determine whether the same results will hold. In addition, we are planning to employ the time-awareness to other models such as RTP to explore whether it continues to support improvement for the open-ended programming environment. Also, this work will be applied to larger groups of students and longer programming tasks, along with integration of more informative features such as intervention and demographic features to develop more robust models. Additionally, we plan to expand our evaluations to longer programs with more complex constructs from both text-based and block-based programming languages.

Acknowledgements: This research was supported by the NSF Grants: EXP: Data-Driven Support for Novice Programmers (1623470), Generalizing Data-Driven Technologies to Improve Individualized STEM Instruction by Intelligent Tutors (2013502), Integrated Data-driven Technologies for Individualized Instruction in STEM Learning Environments(1726550), and CAREER: Improving Adaptive Decision Making in Interactive Learning Environments(1651909).

7. REFERENCES

- [1] R. S. Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *ITS*, pages 406–415, 2008.
- [2] I. Batal, D. Fradkin, J. Harrison, F. Moerchen, and M. Hauskrecht. Mining recent temporal patterns for event detection in multivariate time series data. In *SIGKDD*, pages 280–288. ACM, 2012.
- [3] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou. Patient subtyping via time-aware lstm networks. In *SIGKDD*. ACM, 2017.
- [4] J. E. Beck and J. Mostow. How who should practice: Using learning decomposition to evaluate the efficacy of different types of practice for different types of students. In B. P. Woolf, E. Aïmeur, R. Nkambou, and S. Lajoie, editors, *Intelligent Tutoring Systems*. Springer, 2008.
- [5] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis – a general method for cognitive model evaluation and improvement. In M. Ikeda, K. D. Ashley, and T.-W. Chan, editors, *Intelligent Tutoring Systems*, pages 164–175. Springer, 2006.
- [6] C. Che, C. Xiao, J. Liang, B. Jin, J. Zho, and F. Wang. An rnn architecture with dynamic temporal matching for personalized predictions of parkinson's disease. In *SDM*. SIAM, 2017.
- [7] M. Chi, K. R. Koedinger, G. J. Gordon, P. Jordon, and K. VanLahn. Instructional factors analysis: A cognitive model for multiple instructional interventions. In *EDM*, pages 61–70, 2011.
- [8] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *MLHC*, pages 301–318, 2016.
- [9] M. Cocea and S. Weibelzahl. Can log files analysis estimate learners' level of motivation? In *LWA*, pages

- 32–35. University of Hildesheim, Institute of Computer Science, 2006.
- [10] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMUAI*, 4(4):253–278, 1994.
 - [11] S. R. Eddy. Hidden markov models. *Current opinion in structural biology*, 6(3):361–365, 1996.
 - [12] C. Esteban, O. Staeck, et al. Predicting clinical events by combining static and dynamic information using recurrent neural networks. In *ICHI*, pages 93–101. IEEE, 2016.
 - [13] M. Feng, J. Beck, N. Heffernan, and K. Koedinger. Can an intelligent tutoring system predict math proficiency as well as a standardized test? In *EDM*, pages 107–116, 2008.
 - [14] A. Galyardt and I. Goldin. Recent-performance factors analysis. In J. Stamper, Z. Pardos, M. Mavrikis, and B. McLaren, editors, *Proceedings of the 7th International Conference on Educational Data Mining*, pages 411–412, 2014.
 - [15] D. Garcia, B. Harvey, and T. Barnes. The Beauty and Joy of Computing. *ACM Inroads*, 6(4):71–79, 2015.
 - [16] W. J. González-Espada and D. W. Bullock. Innovative applications of classroom response systems: Investigating students’ item response times in relation to final course grade, gender, general point average, and high school act scores. *Electronic Journal for the Integration of Technology in Education*, 6:97–108, 2007.
 - [17] A. Graves, N. Jaitly, and A.-r. Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *ASRU*, pages 273–278. IEEE, 2013.
 - [18] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
 - [19] X. Jia, A. Khandelwal, G. Nayak, J. Gerber, K. Carlson, P. West, and V. Kumar. Incremental dual-memory lstm in land cover prediction. In *SIGKDD*. ACM, 2017.
 - [20] X. Jia, S. Li, A. Khandelwal, G. Nayak, A. Karpatne, and V. Kumar. Spatial context-aware networks for mining temporal discriminative period in land cover detection. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 513–521. SIAM, 2019.
 - [21] E. Joseph. Engagement tracing: using response times to model student disengagement. volume 125, page 88. IOS Press, 2005.
 - [22] M. Khajah, R. V. Lindsey, and M. C. Mozer. How deep is knowledge tracing? *arXiv preprint arXiv:1604.02416*, 2016.
 - [23] F. Khoshnevisan, J. Ivy, M. Capan, R. Arnold, J. Huddleston, and M. Chi. Recent temporal pattern mining for septic shock early prediction. In *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 229–240. IEEE, 2018.
 - [24] C. Lin and M. Chi. A comparisons of bkt, rnn and lstm for learning gain prediction. In *AIED*, pages 536–539. Springer, 2017.
 - [25] Z. C. Lipton, D. C. Kale, et al. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.
 - [26] M.-T. Luong and C. D. Manning. Stanford neural machine translation systems for spoken language domains. In *IWSLT*, pages 76–79, 2015.
 - [27] Y. Mao, C. Lin, and M. Chi. Deep learning vs. bayesian knowledge tracing: Student models for interventions. *JEDM*, 10(2):28–54, 2018.
 - [28] C. H. McGrath, B. Guerin, E. Harte, M. Frearson, and C. Manville. Learning gain in higher education. *Santa Monica, CA: RAND Corporation*, 2015.
 - [29] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, pages 4694–4702. IEEE, 2015.
 - [30] S. Pandey and G. Karypis. A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*, 2019.
 - [31] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *UMAP*, pages 255–266. Springer, 2010.
 - [32] Z. A. Pardos and N. T. Heffernan. Kt-idem: Introducing item difficulty to the knowledge tracing model. In *UMAP*, pages 243–254. Springer, 2011.
 - [33] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis –a new alternative to knowledge tracing. In *AIED*, pages 531–538, 2009.
 - [34] T. Pham, T. Tran, D. Phung, et al. Deepcare: A deep dynamic memory model for predictive medicine. In *PAKDD*. Springer, 2016.
 - [35] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *NIPS*, pages 505–513, 2015.
 - [36] T. Price, R. Zhi, and T. Barnes. Evaluation of a data-driven feedback algorithm for open-ended programming. *International Educational Data Mining Society*, 2017.
 - [37] T. W. Price, Y. Dong, and D. Lipovac. iSnap: Towards Intelligent Tutoring in Novice Programming Environments. In *Proceedings of the ACM Technical Symposium on Computer Science Education*, pages 483–488, 2017.
 - [38] Y. Qiu, Y. Qi, H. Lu, Z. A. Pardos, and N. T. Heffernan. Does time matter? modeling the effect of time with bayesian knowledge tracing. In *EDM*, pages 139–148, 2011.
 - [39] S. Ritter, A. Joshi, S. Fancsali, and T. Nixon. Predicting standardized test scores from cognitive tutor interactions. In *EDM*, pages 169–176, 2013.
 - [40] D. L. Schnipke and D. J. Scrams. Exploring issues of examinee behavior: Insights gained from response-time analyses. In C. N. Mills, M. T. Potenza, J. J. Fremer, and W. C. Ward, editors, *Computer-based testing: Building the foundation for future assessments*, pages 237–266. Mahwah, NJ, US, 2002. Lawrence Erlbaum Associates Publishers.
 - [41] S. Tang, J. C. Peterson, and Z. A. Pardos. Deep neural networks and how they apply to sequential education data. In *L@S*, pages 321–324. ACM, 2016.
 - [42] K. Tatsuoaka. Rule space: An approach for dealing with misconceptions based on item response theory. *JEM*, 20(4):345–354, 1983.

- [43] R. D. L. V. S. Thomas et al. *Response Times: Their Role in Inferring Elementary Mental Organization: Their Role in Inferring Elementary Mental Organization*. Oxford University Press, USA, 1986.
- [44] Y. Wang and N. T. Heffernan. Leveraging first response time into the knowledge tracing model. *International Educational Data Mining Society*, 2012.
- [45] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham. Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation. *arXiv preprint arXiv:1604.02336*, 2016.
- [46] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, pages 802–810, 2015.
- [47] X. Xiong, S. Zhao, E. Van Inwegen, and J. Beck. Going deeper with deep knowledge tracing. In *EDM*, pages 545–550, 2016.
- [48] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *AIED*, pages 171–180. Springer, 2013.
- [49] R. Zhi, T. W. Price, N. Lytle, Y. Dong, and T. Barnes. Reducing the state space of programming problems through data-driven feature detection. In *EDM (Workshops)*, 2018.

Towards Suggesting Actionable Interventions for Wheel-Spinning Students

Tong Mu
Stanford University
tongm@cs.stanford.edu

Andrea Jetten
War Child Holland
Andrea.Jetten@warchild.nl

Emma Brunskill
Stanford University
ebrun@cs.stanford.edu

ABSTRACT

In some computerized educational systems, there is evidence of students *wheel-spinning*, where a student tries and repeatedly fails at an educational task for learning a skill. This may be particularly concerning in low resource settings. Prior research has focused on predicting and modeling wheel-spinning, but there has been little work on how to best help students stuck in wheel-spinning. We use past student system interaction data and a minimal amount of expert input to automatically inform individualized interventions, without needing experts to label a large dataset of interventions. Our method trains a model to predict wheel-spinning and utilizes a popular tool in interpretable machine learning, Shapley values, to provide individualized credit attribution over the features of the model, including actionable features like possible gaps in prerequisites. In simulation on two different statistical student models, our approach can identify a correct intervention with over 80% accuracy before the simulated student begins the activity they will wheel spin on. In our real dataset we show initial qualitative results that our proposed interventions match what an expert would prescribe.

Keywords

Explainable Machine Learning, Inferring Interventions, Wheel-Spinning, Feature Attribution, Shapley Values

1. INTRODUCTION

Educational technology is increasingly used in a wide array of K-12 settings and some students struggle. Beck et al. [6] coined the term “wheel-spinning” to denote students that were repeatedly trying, and failing, to successfully complete a specific skill after many attempts in an intelligent tutoring system. They additionally found it was a significant issue in two popular computerized educational systems. Such long repeated failures are likely to be an inefficient use of time for students, and may additionally contribute to lack of motivation for future learning.

Although expert human instructors are often very good at diagnosing and assisting students who are stuck, it is time consuming for both the instructors and the students waiting for the instructor’s intervention. Additionally many educational settings lack a sufficient number of expert teachers. Our research is particularly motivated by a collaboration with the non-profit War Child Holland whose program Can’t Wait to Learn (CWTL) provides self-paced educational software on tablets primarily to children in or coming from conflict-affected regions. In such settings, a limited number of teachers must often address the learning needs of a large number of students with a wide variety of educational backgrounds. To give a specific example, in the classes in Uganda the program is implemented in, the average class size is 114 students per teacher. Additionally for some population of students where education is especially hard to access, the program is run by facilitators who do not have the same expertise as instructors to provide learning support for individuals. Methods that can automatically identify individualized interventions, such as having the student practice an activity to review a prerequisite skill, to help wheel-spinning students could be greatly beneficial for students and teachers. However, since the term was coined, there has been much work for modelling and predicting wheel-spinning [11, 12, 16], but little work in developing interventions.

There are many possible reasons a student may wheel-spin, including lack of required prior knowledge, a long gap in learning of the material, or an ineffective educational activity. One approach could be to have experts label a large dataset with expert prescribed interventions and train a model to predict those interventions. However in many cases the time necessary to label such a dataset can be infeasibly large. For example, in our real world dataset a domain expert needed 30 minutes to label the 6 wheel-spinning cases we use for a qualitative evaluation. This would translate to 120 expert hours to label our whole dataset of more than 2000 wheel-spinning cases.

In this work we present a method to automatically predict when an intervention could be helpful and which intervention to give. Our method uses prior student system log data and only requires a few hours of expert input. Our method takes as input a set of features, a subset of which are actionable and correspond to a concrete intervention (for example, the feature “prerequisite performance” could correspond to an intervention of reviewing that prerequisite). We then use featurized past student data to train a model

Tong Mu, Andrea Jetten and Emma Brunskill "Towards Suggesting Actionable Interventions for Wheel Spinning Students" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 183 - 193

to predict wheel-spinning. With the prediction model, we use methods from explainable machine for providing feature credit attribution for the prediction of individual datapoints, specifically Shapley values [19], to determine which actionable feature contributed most to the prediction and suggest an intervention.

We evaluate the ability of our method to suggest correct interventions through simulation and through a qualitative study with our real data. Evaluating if our method is impactful will eventually require experimental studies. The costs of an experiment are high in our situation where this educational technology is being used by children in conflict-affected areas and who may be in remote villages without internet. Before embarking on such an effort, in this work we first assess the potential benefits and performance of our method. In simulation studies, we simulated students using two different student models, both based on the popular Bayesian Knowledge Tracing (BKT) [10] student model. In both of our simulations, our method can prescribe a correct action (a helpful intervention or correctly identifying no intervention is needed) with high accuracy before attempting an activity. This accuracy can be improved if the prediction is made at a later attempt. In an initial qualitative assessment in our real world CWTL setting we show our method's explanations are consistent with what an expert would prescribe in a majority of the cases. In the other cases the method did not have access to key data used by the expert, suggesting our method is able to identify correct interventions over correctly defined inputs.

Our method is, to our knowledge, one of the first works for both addressing automatically identifying interventions for wheel-spinning students and using interpretable machine learning in educational technologies. These results suggest that our method can help inform interventions, whether for carefully designed human-in-the-loop systems (such as only informing the teacher if confident the teacher is the best source) or for automated systems (jumping back to practice an earlier skill), and may help further adaptive automated systems for effective, efficient and engaging education.

2. RELATED WORKS

2.1 Wheel-Spinning

The term *wheel-spinning* was first coined by Beck et. al [6] where they examine its prevalence in two educational systems. Gong et al.[11] further explored models to predict wheel-spinning. Beck et al [5] found it applied to students in non-western societies as well. They also examined the influence of affective factors, and found it correlated with gaming the system. Matsuda et al. [16] examined using neural networks together with the BKT model [10] to predict wheel-spinning using only past student performance information. Kai et al. [12] investigate using decision trees to distinguish between productive persistence and wheel-spinning. Zhang et al. [24] make a comparison over many methods for detecting wheel-spinning. Wan et. al [23] take a step in modeling with actionable results by examining the effects of using prerequisite performance as features. They modeled wheel-spinning using both the average prerequisite performance and the weakest prerequisite and found that prerequisite knowledge was a reliable predictor of wheel-spinning and slightly improved model performance. In our work we pro-

Feature	Abbrev	Value	Feature	Abbrev	Value
Activity ID	ID	31	# Attempts Prerequisite 1	P1	High
Time since last played	T	25	# Attempts Prerequisite 2	P2	High
# Attempts Prerequisite 1	P1	7	# Attempts Prerequisite 3	P3	High
# Attempts Prerequisite 2	P2	1			

Wheel Spun?	y	Yes	Wheel Spun?	y	Yes
-------------	---	-----	-------------	---	-----

(a) Example 1: Fake simplified (b) Example 2: Simple Binary datapoint inspired by CWTL Example

Figure 1: Simulated Student Setting

pose a method to not only predict wheel-spinning, but also give suggestion of a possible intervention. We achieve this by designing our features to be actionable, such as incorporating performance on all prerequisites as separate features, with methods from explainable machine learning.

2.2 Explainable Machine Learning

Explainable Machine Learning is a rapidly growing popular field in the machine learning community. One subfield is the study of feature attribution which are methods that return how much each feature contributed to the total prediction of a datapoint in a machine learning model. In our work we use Shapely values [19], and the python implementation SHAP [13, 14] package to inform interventions. Shapley values are a method originating in game theory for fairly allocating a payout between participants. It has recently found popularity in explainable machine learning to calculate feature attributions. Shapley values have been used widely both within and outside of machine learning, including in medical applications [15], social network node analysis [17], and studying carbon emission quotas in China [25]. To our knowledge this is one of the first works on using Shapley values and explainable machine learning methods for educational technologies.

3. METHODS

In this section we present an algorithm to help students likely to wheel spin by suggesting actionable interventions. Our goal is to provide a method for using past student log data to predict when and which intervention a student will need to prevent wheel-spinning. We would also like to minimize interruptions to student-activity pairs who do not wheel spin. Similar to prior work [6] we define wheel-spinning as when a student consecutively fails an educational activity more than a threshold number of times. We will refer to the student-activity pair of the i^{th} student working on the j^{th} activity as $pair_{ij}$. To achieve our goal, for every $pair_{ij}$, our algorithm uses a 2 level decision process shown in Algorithm 1.

We first train a machine learning model using an existing dataset of student log data to predict wheel-spinning. Our overall algorithm (Algorithm 1) is compatible with any machine learning model that outputs probabilities of wheel-spinning given an input set of student features. In our work we use the popular gradient boosting method XGBoost [9]. When a student is using the educational program, given their current state the algorithm uses the trained model to predict if a student-activity pair will result in wheel-spinning. We define the number of failed attempts a student

makes before we decide to possibly suggest an intervention as n . If the student has reached the n^{th} attempt on the current item our method uses the wheel-spinning model to predict if wheel-spinning will occur. If the output probability of wheel-spinning of the model is greater than a threshold, p , the algorithm will then propose a potential intervention. n and p are hyperparameters, and we provide further discussion on their effect in Sections 4.5 and 6.

Interventions are proposed using a method of feature attribution from explainable machine learning, Shapley Values [19] (described in more detail in Section 3.1). We use Shapley values to assign a contribution value to each feature used in the wheel-spinning prediction model. A subset of these features are designed to be actionable and correspond to an intervention. For example, Figure 1a shows example feature values of a fake datapoint for a student-activity pair inspired by CWTL. An example of an actionable feature in this fake datapoint is number of attempts required on a prerequisite skill. If assigned a high positive attribution value, it would suggest the student needs more practice on that prerequisite. Our method identifies the actionable feature with the highest Shapley value and suggests the corresponding intervention to give to the student. Non-actionable features that do not correspond to an intervention but increase prediction accuracy are also included.

There are a few places that require expert input, for example choosing hyperparameters n and p and designing the features and interventions. For experiments with our real world dataset, we worked together with a domain expert to create actionable features and corresponding interventions.

3.1 Background on Shapley Values

In this section we provide some background on the calculation and properties of Shapley Values [19] which is used in our method to provide feature attribution for the wheel-spinning prediction of individual datapoints. Shapley values originated in game theory and in the context of explainable machine learning, provide an attribution for how much each feature contributes to the total prediction of a datapoint. To give an example, consider a setting where we are predicting wheel-spinning using features in our dataset. We will refer to this setting as example setting 1. The datapoint in Figure 1a gives an example datapoint in this setting. Assume the mean prediction of the wheel-spinning model over all the datapoints in this example is 0.5, and for this datapoint the model predicts a probability of 0.8, which is +0.3 from the mean. The Shapley values for each feature give the contribution of each feature to this difference from the mean where the sum of contributions over all features must be +0.3. For example the features ID , T , $P2$ could all be attributed -0.1 and the feature $P1$ could be attributed +0.6. This attribution would suggest the value of the number of attempts on prerequisite 1 is likely to be responsible for the increased probability of wheel-spinning over the average wheel-spinning prediction.

Shapley values is the only method for attribution that satisfies the following desirable properties which together are the definition of a fair attribution [19]: symmetry (two features that contribute equally will have the same value), dummy (a feature that does not change the prediction has a value of

0), and additivity (if the prediction model is the sum of multiple models, the value of a feature in the prediction model is the sum of all values over the individual models). To give intuition of why the symmetry and dummy properties are desirable in this context, consider a second, simpler example setting, example setting 2, where we are also predicting wheel-spinning but all inputs and outputs are binary. In this setting the wheel-spinning prediction is for one activity that is thought to have three prerequisites ($P1$, $P2$, $P3$). Figure 1b gives an example datapoint in this setting. Consider the case where two prerequisites, $P1$ and $P2$, are equally important and $P3$ was incorrectly labeled as a prerequisite and its value never influences the prediction of the model. Because $P1$ and $P2$ are equally important and for the datapoint in figure 1b both their values are high, we would like them to have equal attribution, or to satisfy the “symmetry” property. Additionally, because $P3$ was incorrectly labelled as a prerequisite we would want it to be given 0 attribution regardless of its value, or to satisfy the “dummy” property.

We now describe formally how to calculate Shapley values. Let \mathcal{F} denote the set of features and X denote the dataset. One example of a datapoint in X from example setting 1 is the example datapoint Figure 1a, which we will refer to as x_i . In this example $\mathcal{F} = \{ID, T, P1, P2\}$. Also assume there is a function V where $V(x_i)$ is the predicted value on datapoint x_i . Let $SHAP(x_i, f_j)$ be x_i ’s Shapley value for feature f_j . In our example, $V(x_i)$ would output the probability of x_i wheel-spinning. For a subset of features s , ($s \subseteq \mathcal{F}$) we define a fake datapoint, $x_{i,s}$, as a datapoint that only includes the the values of x_i for the features in s . In our example, one potential s could be $\{T, P1\}$, and the corresponding $x_{i,s} = [T : 25, P1 : 7]$. For a feature f_j , we define a coalition of features, F , as a subset of \mathcal{F} that does not include f_j . We define \mathcal{C} as the set of all unique coalitions for f_j and let F_k denote the k^{th} coalition in this set. Let the contribution of f_j in coalition F_k to the prediction of x_i be the difference in prediction of the datapoint without f_j and the datapoint with f_j included, or $V(x_{i, F_k \cup f_j}) - V(x_{i, F_k})$. In our example, the $s = \{T, P1\}$ is a coalition of features for $f_j = ID$ as it does not include f_j . If the probability of wheel-spinning on $x_{i,s}$ ($V([T : 25, P1 : 7])$) is -0.1 and $V(x_{i, F_k \cup f_j}) = V([ID : 31, T : 25, P1 : 7]) = -0.2$. Then the difference $V(x_{i, F_k \cup f_j}) - V(x_{i, F_k}) = -0.1$

The Shapley value is then the expected contribution of f_j averaged over all coalitions:

$$\begin{aligned} SHAP(x_i, f_j) &= \mathbb{E}_{\mathcal{C}}[V(x_{i, F \cup f_j}) - V(x_{i, F})] \\ &= \sum_{F_k \in \mathcal{C}} \frac{|F_k|!(|\mathcal{F}| - 1 - |F_k|)!}{|\mathcal{F}|!} (V(x_{i, F_k \cup f_j}) - V(x_{i, F_k})) \end{aligned} \quad (1)$$

$$(2)$$

Going back to example setting 2 (Figure 1b), we see once either $P1$ or $P2$ enters a coalition that does not contain either of them (so $\{\emptyset\}$ or $\{P3\}$) the prediction increases from zero to one and will not increase further when the other enters. Because we average across all coalitions and in half of the coalitions, $P1$ will occur before $P2$ and in the other half $P2$ will occur before $P1$, the symmetry property will be

satisfied and $P1$ and $P2$ will be given equal attribution. We can also see that if $P3$ does not change the prediction value in any coalition, it will be given zero attribution, satisfying the dummy property.

In the machine learning case, we would like to use a machine learning model M as the function that assigns a predicted value to x_i . Because a machine learning model requires a datapoint to have values for all features, we must approximate $V(x_{i,F_k})$ using other datapoints. Let x_l be a randomly sampled real datapoint from the dataset that is not x_i . We define a fake datapoint $x_{i,F_k,l}$ as a hybrid datapoint that contains the feature values of x_i for the features in F_k and the feature values of x_l for the features not in F_k . In our running example, $x_{i,F_k,l} = [ID: x_{l,ID}, T: 7, P1:6, P2:x_{l,P2}]$. $M(x_{i,F_k,l})$ is then used to approximate $V(x_{i,F_k})$.

Shapley values require summing over all possible coalitions and are very computationally expensive. There are algorithms that compute an approximate solution through sampling such as the method proposed by Vstrumbel et al [20]. In our case, we use an implementation, TreeSHAP [13, 14], designed to efficiently and quickly calculate exactly Shapley Values for decision tree based models.

Algorithm 1: Suggest Intervention for $Pair_{ij}$

Input : Dataset of Preexisting Log Files (\mathcal{D}), Set of Actionable Features (\mathcal{F}_a), Set of other features (\mathcal{F}_o), Mapping of Actionable Features to Interventions (**GetIntervention**), $student_i$ log file (L_i) at n^{th} attempt on $problem_j$, wheel-spinning Model Output Probability Threshold (p)

Output: Suggested Intervention for $Pair_{ij}$

// We abbreviate wheel-spinning as WS

$WSModel = \text{TrainModel}(\mathcal{D}, \{\mathcal{F}_a, \mathcal{F}_o\}, n)$

$X_i = \text{GetCurrentFeatures}(L_i, \{\mathcal{F}_a, \mathcal{F}_o\})$

$q = WSModel.predict(X_i)$

if $q > p$ **then**

$\{SHAP_a, SHAP_o\} = \text{ComputeShapley}(X_i, WSModel, \{\mathcal{F}_a, \mathcal{F}_o\})$ // Section 3.1

$MaxFeature = \text{argmax}_{f_a} SHAP_a$

$Intervention = \text{GetIntervention}(MaxFeature)$

else

$Intervention = \text{Don't Intervene}$

end

3.2 Baselines

We compare to two baselines and, in this section, include discussion for building intuition for which situations our proposed method could outperform the baselines.

Baseline 1 - Overall Feature Importance (FI): Because we are using a decision tree based method to predict wheel-spinning, overall feature importances are calculated automatically. Therefore, we can consider a method that when a student-item pair is predicted to wheel spin, choose the intervention suggested by the feature with the highest overall feature importance. This method requires less compute as it does not require an additional step of calculating individualized feature attributions. Conceptually, this method will perform equivalently as our proposed method when there is

a single cause for wheel-spinning. However in cases where there can be many potential causes (for example, some student-item wheel-spinning is due to forgetting effects from long durations between learning while others are due to unmastered prerequisites), then this baseline, which will only select the single, most predictive cause for all students, will perform poorly. In this respect, this baseline has parallels to a baseline which predicts the majority class. Note that we do not compare to a baseline that predicts the majority class because we are considering a setting where we do not have any labels for wheel-spinning causes. Consequently our method has no way of discerning what the majority cause is. The goal of our work instead is investigating the effectiveness of feature attribution methods to identify causes.

Baseline 2 - Logistic Regression (LR): Linear models such as logistic regression are a computationally efficient subset of our method as they, by nature and without needing additional calculation, have feature credit attribution for the predictions of individual datapoints. They can potentially work well in cases where a linear relationship can accurately model the relation between features and wheel-spinning. However in many domains, such as CWTL, non-linear models for the wheel-spinning prediction can achieve better performance (shown in Section 5.4). Therefore in this work we focus on a method that can work with non-linear models and we treat linear models as a baseline.

4. SIMULATIONS

We assess the performance of our method in simulation where we can create true causes of wheel-spinning, which we define as needing 10 or more attempts on one educational activity to match both prior work [11, 12] and evidence from the CWTL data.

We simulate students using two different student models both based on the Bayesian Knowledge Tracing (BKT) model [10]. The BKT model is a two state Hidden Markov Model (HMM) and is a popular model of student learning that has been shown to be successful for various applications in the educational technology literature (for example Corbett et al. [10]). The model has two hidden states, mastered or not mastered, and two observed states, correct or incorrect. From the mastered state of a skill, the student will answer an educational activity involving that skill correctly unless they slip and answer incorrectly with a probability of slip ($P(s)$). From the unmastered state of that skill, a student will answer a problem involving the skill incorrectly, unless they guess correctly with a probability of guess ($P(G)$). Everytime the student is presented a practice opportunity for an unmastered skill, they have a probability of transitioning ($P(T)$) to the mastered state for the skill. We make modifications to the BKT model to match aspects of the CWTL domain that may also occur in other domains. In our simulations we specifically consider a situation where a student may have been moved on too fast because they passed a prerequisite by guessing. This is because the corresponding intervention of reviewing the relevant prerequisite could be automated and is a key feature we are trying to achieve in the CWTL setting.

4.1 Simulated Curriculum

In Figure 2a we illustrate our simulated sequence of educational activities as well as the prerequisite structure between them. In this setting we consider each activity as corresponding to a unique skill. Skills build on each other in the way shown in the prerequisite graph. To mimic the CWTL curriculum, simulated students are presented educational activities in order starting at A1. They are repeatedly presented an educational activity (for example A1) until they succeed and are moved onto the next activity (in our example, A2).

We note that while our analysis and results are in a setting where the curriculum is linear, our method does not rely on this setting and can be applied more generally to different types of ordering constraints over educational activities.

4.2 Student Model 1

In our first student model, we make two modifications to the BKT model to reflect behaviors that occur in our domain and in other domains. In CWTL, each activity involves answering a certain percentage of multiple choice questions relating to the target skill of the activity correctly. In this setting, the probability of guess starts low, however questions are reused between activity instances so the probability of guess increases with attempts as students may start to memorize answers. This effect can also occur in other domains where questions are reused. To mimic this effect in simulation, we start the probability of guess at a low base value $P(G)$ and with every attempted answer by the student, we increase it in such a way that at the n^{th} attempt of the student on the problem the probability of guess, $P_n(G)$, is $P_n(G) = 1 - (1 - P(G))^n$. We use this function as it monotonically increases to its limit of 1.

Our second modification is, for skills involving prerequisites (A4 and A5), we enforce the prerequisite structure by defining a new transition probability for when the prerequisites are not mastered, $P_{unmastered}(T)$. In all our simulations this was set to zero however this probability can also be set to a small non-zero probability with similar results. This is to reflect the difficulty of learning complex combinatorial skills without mastering the prerequisites.

4.2.1 Data Generation

In our simulations, we show our method is able to correctly distinguish when and which prerequisite should be reviewed. We consider the whole population comprised equally of two different populations of students. Students of student population 1 finds all skills “easy” to master and has high transition probabilities for all skills. Students of student population 2 finds one of the prerequisite skills (A1, A2, or A3) “hard” to master and has low transition probabilities for that skill. For this student model, we are able to control which prerequisite students of student population 2 may not master by setting that prerequisite as “hard”. Additionally we can examine the performance of our method at suggesting interventions in a heterogeneous population.

We report results from one set of parameters with the transition dynamics described in Table 1. Notice $P(G)$ is lower for A4 and A5 to reflect the complexity of those two questions over A1, A2, and A3. We generate both our training and test sets by simulating 1000 student trajectories, 500 from

Table 1: Parameters for Student Model 1

P(T) “easy”	P(T) “hard”	P(G) (A1,A2,A3)	P(G) (A4,A5)	P(S)	“hard” skill
0.5	0.01	0.01	0.005	0	A2

Table 2: Parameters for Student Model 2

P(T)	P(D)	P(G) (A1,A2,A3)	P(G) (A4,A5)	P(S)
0.5	0.1	0.01	0.005	0

each population. For these simulation parameters, initially $P_{easy}(T)$ is higher than $P_n(G)$ and if a student needs a low number of attempts on a prerequisite, they are most likely part of student population 1 and have mastered the prerequisite. If a student needs a higher number of attempts on a prerequisite, then they are most likely in student population 2 and they may either have mastered the prerequisite or passed through guessing and need to repractice the prerequisite. Decreasing the value of $P_{easy}(T)$ or $P(G)$ can increase the strength of this correlation between attempts and mastery and allow model accuracy to increase. Similarly, increasing these parameters, or increasing $P_{unmastered}(T)$ can decrease accuracy.

4.3 Student Model 2

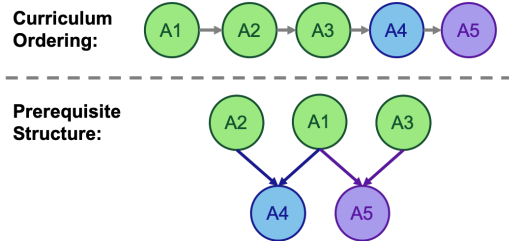
We designed our second simulated model to account for student engagement and simulate disengagement and wheel-spinning behavior. We did so based on expert insights, and findings from prior literature on boredom and disengagement in tutoring systems. A figure illustrating this modified model is shown in Figure 2b.

In this model we make an additional modification on Student Model 1 by splitting the “Not Mastered” state into two states: “Engaged” and “Disengaged”. Each student for each activity starts in the Engaged state. In the Engaged state the student is open to learning and can transition to the “Mastered” state with probability $P(T)$. However with each failed activity attempt, on the n^{th} attempt they can also transition to the “Disengaged” state with probability $P_n(D)$. This probability of disengagement starts at 0 and is parametrized by a base value of $P(D)$. It increases monotonically in the same way the probability of guess does, to eventually reach 1: $P_n(D) = 1 - (1 - P(D))^n$. Once in the disengaged state for a skill, the student can transition out of it with probability $P(E)$. In our simulations we set $P(E)$ to 0 however it can also be set to a small non-zero probability with similar results.

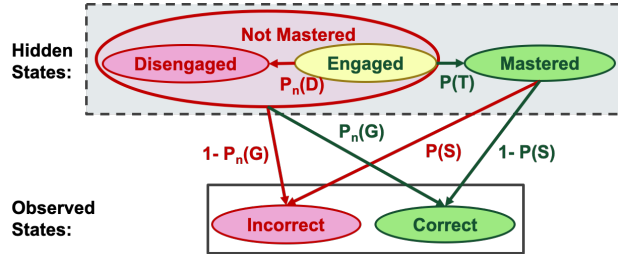
We make these modifications to reflect points from (1) prior literature and from domain expert insights that suggests repetitive tasks can lead to boredom [22, 8], (2) literature suggesting boredom can lead to disengagement which results in gaming behavior (such as random guessing) [3, 2, 4, 1, 11] as opposed to productive learning (3) literature suggesting disengagement and boredom are affective states that persist and are hard to transition out of [4, 1, 18].

4.3.1 Data Generation

We generate both our training and test set by simulating and generating 1000 student trajectories. Parameters used



(a) Simulated Curriculum



(b) Diagram of the modified BKT of Student Model 2

Figure 2: Simulated Student Setting

to generate the results are given in Table 2. For these parameters because $P(T)$ is initially much higher than $P_n(G)$ and $P_n(D)$, if a student needs a low number of attempts, they most likely mastered the activity. If they need a large number of attempts, they most likely became disengaged and guessed correctly. In these simulations, the correlation between attempts and mastery can be increased by increasing $P(T)$ or decreasing either $P(G)$ or $P(D)$. Similarly changing the parameters in the opposite direction or increasing $P_{unmastered}(T)$ or $P(E)$ can decrease accuracy.

4.4 Features

We train our model to predict wheel-spinning on the later skills, A4 and A5, and automatically suggest interventions in the form of if and which prerequisite to review. In both of the student models, needing a higher number of attempts on an activity is positively correlated with a skill not being learned. With this in mind we use the following three features and corresponding interventions: (1) Activity identity (A4 or A5): If assigned a high contribution, the corresponding intervention could be redesigning the level. (2): Number of attempts on the most recent prerequisite as defined by the prerequisite graph. The corresponding intervention would be to have the student review that activity. (3): Number of attempts on the second most recent prerequisite.

4.5 Results

4.5.1 Evaluation Metrics

To evaluate the accuracy of our method, we consider the frequency with which the method predicts a correct action, which includes correctly deciding to not intervene and correctly suggesting a correct intervention. We refer to student-problem pairs that would lead to wheel-spinning if no intervention is given as a wheel-spinning pair and student-problem pairs that would not wheel spin if no intervention is given as non-wheel-spinning pairs. Across all student-problem pairs, we define four counts:

1. Correct-Pairs_No-Intervention (CP_NI): the number of student-problems where the model correctly suggests no intervention)
2. Correct-Pairs_Intervention (CP_I): model correctly suggests the right intervention

Student Model	n	Method	Accuracy	Precision	Recall	F1	AUC
1	0	XGB	88%	0.68	0.72	0.70	0.89
		LR	86%	0.71	0.50	0.59	0.89
	5	XGB	94%	0.79	0.93	0.85	0.97
2	0	XGB	83%	0.75	0.58	0.65	0.79
		LR	80%	0.75	0.44	0.55	0.79
	5	XGB	93%	0.81	0.99	0.90	0.96

Table 3: Simulation wheel-spinning prediction results averaged over 200 simulations. XGB refers to XGBoost, LR refers to the Logistic Regression baseline. At $n=5$ attempts the performance of XGB and LR are very similar so only the XGB results are included. At $n=0$ attempts, XGB has higher Accuracy and F1 than LR.

3. Missed-Pairs (MP): model either suggests an incorrect intervention or incorrectly does not suggest an intervention)
4. Interrupted-Pairs (IP): model incorrectly suggests giving an intervention when it is unneeded, or suggests the wrong intervention

Note that IP and MP both include students that were wheel-spinning but the model suggests the wrong intervention since such students are both not helped (“missed”) and would be asked to do something not useful (“interrupted”). Additionally, we classify wheel-spinning students who mastered both prerequisites and were still jumped back to a prerequisite as CP_I as insights from our domain expert suggests that jumping back when a student is wheel-spinning and possibly disengaged can be a helpful intervention.

Let S be the total number of student-problem pairs and define *accuracy* as the total percentage of student problem pairs that were given a correct intervention ($= \frac{CP_NI + CP_I}{S}$); *miss rate* as the percentage of wheel-spinning instances that were not identified or which were proposed the incorrect intervention ($= \frac{MP}{CP_I + MS}$), and the *interrupted rate* as the percentage of Interrupted Pairs out of all student-problem pairs that did not need an intervention ($= \frac{IP}{CP_NI + IP}$).

4.5.2 Results

For all results, we averaged over $N=200$ simulations by repeating 200 times the data generation procedure outlined in Sections 4.2.1 and 4.3.1. With this size of N , the standard deviation for all results reported in this section is less than 0.005 (for results reported in percentages, less than 0.5%).

Student Model	n	Method	Accuracy	Miss Rate	Interrupted Rate
1	0	Ours	88%	28%	8%
		LR	86%	50%	5%
	5	Ours	92%	14%	8%
		LR	92%	14%	8%
2	0	Ours	83%	42%	8%
		LR	80%	56%	5%
		FI	75%	68%	16%
	5	Ours	92%	4%	10%
		LR	86%	25%	17%
		FI	84%	34%	19%

Table 4: Simulation intervention suggestion results, averaged over 200 simulations. Ours refer to our proposed method, LR refers to the Logistic Regression baseline, FI refers to the overall XGBoost feature importance baseline. Notice the FI baseline was not included for Student Model 1 because in that simulation, there was only one cause of wheel-spinning (Prerequisite 2) so FI is exactly equivalent to our method.

We report the results of the XGBoost and Logistic Regression (baseline) models for predicting wheel-spinning in Table 3. For lower values of n , XGBoost can achieve higher accuracy and F1 when predicting wheel-spinning. As n increases, the dataset becomes heavily skewed towards data-points with wheel-spinning as well as students needing less than n attempts correctly automatically labelled as no-wheel-spinning, resulting in both methods achieving high accuracy.

We report the results of our method for identifying interventions for both student models in Table 4 when making the prediction at 0 attempts and 5 attempts ($n = 0$ and $n = 5$). The probability threshold of the wheel-spinning model over which we suggest an intervention (p) was set to 0.5 for both. Our approach achieves high accuracy for both student models even when making early predictions before the student begins an activity (0th attempt). Additionally our method is mostly able to do better than the Logistic Regression baseline (LR). For Student Model 1, because there is only one cause of wheel-spinning the prescriptions of the XGBoost Overall Feature Importance Baseline (FI) was exactly the same as our method. However in Student Model 2 where there is more than one cause of wheel-spinning, our method performs much better.

Due to the fact that students are modelled stochastically, we are not able to achieve 100% accuracy as the correlation between number of attempts on a problem and problem mastery is not perfect. However we can increase the accuracy by making the prediction at a later number of attempts as shown in Table 4 when the intervention prediction made at the fifth attempt ($n = 5$). Our accuracy for both student models increases and the miss rate for both decreases. As we increase the attempt number at which we consider providing an intervention, all the student problem pairs that resulted in less than 5 attempts were correctly not intervened upon and automatically categorized as CP_NI. We provide further discussion of this hyperparameter and the p hyperparameter in the Discussion (Section 6).

5. CAN'T WAIT TO LEARN

Our method was motivated by our collaboration with the Can't Wait to Learn (CWTL) program of War Child Holland. CWTL is a tablet based, curriculum aligned, self-paced, autonomous learning program that aims to teach basic numeracy and literacy skills to children in conflict-affected settings who are facing challenges in accessing quality education. The program is delivered on a tablet and targets learning objectives from grade 1-3. Based on the context, the program can be used as a standalone or a supplemental educational program. CWTL is currently rolled out in Sudan, Lebanon, Jordan, Chad, Bangladesh and Uganda. Prior studies found the program was able to result in increased psychological well-being as well as positive learning outcomes in multiple countries [7, 21].

5.1 Game Mechanics

For our application we focus on the English reading program in Uganda where we notice a high amount of wheel-spinning. In classrooms utilizing the program, the instructor to student ratio is large, with class sizes of 114 students per teacher on average. The game takes place in the game world shown in the left panel of Figure 3a. In the game, the student is a member of a Ugandan village and the overarching narrative of the game is to help each village member achieve their goals by playing educational mini-games. The educational mini-games (Figure 3a right panels give two examples) and the instructional videos explaining concepts, such as letters or more complex vowel sounds, form the main educational mechanism. Each educational activity in the program is a specific instance of a mini-game and the curriculum is a fixed linear curriculum of a sequence of these educational activities. For example, in the mini-game at the top right of Figure 3a, the goal concept is learning to combine sounds of words beginning with “o”. In the specific practice question shown of this mini-game, students first tap the blue buttons to listen to the sounds the “o” and “ff” components of the word make separately. To answer the question correctly, they must then tap the correct picture describing the complete word (“off”). To succeed on the activity students must answer 8 out of 10 instances of this question correctly as described by the green the orange circles displayed at the top. Students practice each activity repeatedly until they achieve this success criteria. When a student succeeds at an educational activity they are progressed to the next activity in the curriculum.

5.2 Wheel-Spinning Details

In analyzing the data, we find that 2.4% of student-problem pairs exhibit wheel-spinning. This is lower than in other systems because there exist easier activities for entertainment, engagement, morale, and for gaining initial familiarity with a new concept without too much cognitive overload. Wheel-spinning is still a problem as we find that 51% of students wheel spin at least once. The bottom plot of Figure 3b shows time played compared with the last activity reached. The students who are below the curve whom we would like to help are circled in orange.

To determine the threshold of attempts to define wheel-spinning, we examine plots of student attempts on the game they are currently playing at the end of the most recent log file. If students are stuck on an activity, they are spending more time on it and have a higher probability of being

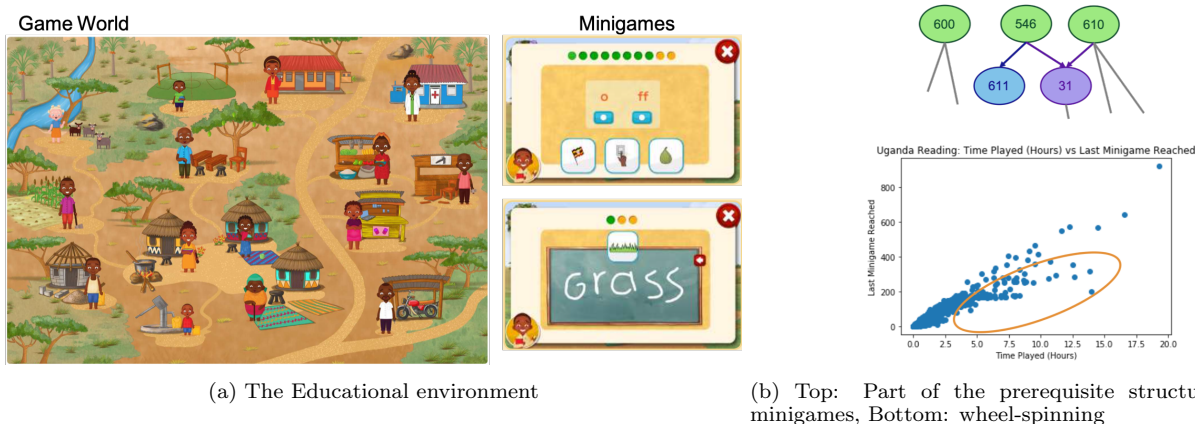


Figure 3

on that activity when the playing session ends. Therefore the activity the student is currently playing at the time the log file was accessed is correlated with activities students are wheel-spinning on. We compared the distribution of attempts of the problem students are currently on to the distribution of attempts on the activities they played 1 or 5 activities ago, which are less correlated with wheel-spinning. We find a non-negligible percentage of students need 10 or more attempts on the current activity they are playing (27%) while few students require 10 or more attempts on activities played 1 activities (6.8%) or 5 activities (5.9%) ago. We therefore defined wheel-spinning as failing 10 or more attempts on an activity.

5.3 Model

In our model we used the following actionable features and describe the corresponding intervention. We highlight the actionable features that allow for in game interventions in **bold**. These are especially helpful in our domain where student to teacher ratios may be large. We also provide an example of a non-actionable feature¹:

(1) **Last Played**: If there has been a long duration since the student last played, the intervention is to diagnose and have them review what they forgotten. (2) **Number of attempts on the Prerequisite 1, 2 and 3 Prerequisites ago**: A small portion of the prerequisite structure is shown in the top image of Figure 3b. These features use the prerequisite graph to find the last, second to last, and third to last prerequisite in the curriculum. These features in the CWTL domain can be evidence that a student did not master the corresponding prerequisite. The intervention is to have the student practice the prerequisite. (3) **Mini-game Type**: Allows the model to identify if a mini-game should be redesigned. (4) **Number of attempts on the first video**: To pass any video, a student only needs to watch it completely.

¹We also included other non-actionable features to reduce confounding and improve prediction accuracy which we omit in sake of clarity and brevity. Some examples of other non-actionable features included were the number of times mini-game was seen before, the Learning Level, which gives a rough location of where the student is in the curriculum, as well as other features helpful for distinguishing current student location in curriculum.

The number of attempts on the first video can be an indicator of low technological fluency. The intervention is to have a notification that encourages them to ask a teacher or a peer for help. (5) **First Time Mini-game Type Seen?**: Students generally will need more attempts the first time they experience a mini-game. So this feature, while not actionable, allows the model to make more accurate predictions.

5.4 Results

We first examine the accuracy of our model at predicting wheel-spinning. We used data from 1170 students. Students were assigned randomly to the training and test set with 80%, or 943, students assigned to the training set. The students completed 60 activities on average. There were a total of 55,035 student-activity pair datapoints in the training set with 1,294 of them as wheel-spinning (2.4%). There were a total of 15,004 datapoints in the test set with 322 of them as wheel-spinning (2.2%). These datapoints were all used in the $n = 0$ condition. Considering only the student-activity pairs that required 5 or more attempts ($n = 5$), the training set had 2568 datapoints (50% wheel-spinning - there were still 1,294 wheel-spinning datapoints since only datapoints with less than 5 attempts were removed) and the test set had 664 datapoint (48% wheel-spinning). At $n = 9$, the training set had 1454 datapoints (89% wheel-spinning) and the test set had 365 datapoint (88% wheel-spinning).

As shown in Table 5, while our accuracy is quite high, due to the class imbalance, precision, recall, and F1 are low. We tried a variety of different models such as CART decision trees and Random Forests and we found the model we used, XGBoost, to do the best by a slight margin over Random Forests and significantly over CART. We additionally report results for Logistic regression to show that for lower values of n , it is not able to achieve the same accuracy as XGBoost. As with the simulations, as the value of n increases, the accuracy difference of the two models on predicting wheel-spinning decreases as both models achieve high accuracy at higher values of n . This is due both to a higher balance of wheel-spinning datapoints in the dataset and automatically correctly predicting not-wheel-spinning on students who needed less than n attempts. However this increased accuracy at higher values of n is at the expense of allowing some of the students who will eventually wheel-spin

n	Method	Accuracy	Precision	Recall	F1	AUC
0	XGB	93%	0.21	0.60	0.31	0.91
	LR	88%	0.12	0.60	0.19	0.86
5	XGB	98%	0.60	0.60	0.60	0.99
	LR	98%	0.53	0.58	0.55	0.99
9	XGB	99.7%	0.90	1	0.94	0.999
	LR	99.7%	0.88	1	0.94	0.99

Table 5: Wheel-Spinning Prediction: XGB refers to XGBoost, LR refers to the logistic regression baseline. LR has worse predictive accuracy and a lower F1 score than XGB when the prediction is made at lower values of n .

on a problem still spend multiple attempts on the problem. To deploy a system we would work with a domain expert to decide the n that would be best.

To verify the method, we compare our method’s predictions to those an expert would prescribe. To obtain the expert prescription, we blinded the domain expert author of this paper, by showing them the cases and asking for their prescriptions before sharing with them the details or results of the model. To generate the test cases, we randomly sampled true wheel-spinning student-problem pairs of that were also predicted as wheel-spinning by the model. To get diverse cases, sampling was done by throwing out newly sampled cases that were very similar to two or more previously selected cases, until we had 6 cases total. For purposes of making a comparison, we made a list of possible causes and interventions for the domain expert to choose from, including a none-of-the above choice. In our model, some features allow for immediate actions (reviewing a prerequisite problem) while others do not (redesigning an educational activity). The immediately actionable features are much easier to intervene on and based on our expertise gained, are much more favorable to an expert or instructor. To reflect this, we made the decision (before discussing the methods and giving the examples to the expert) to choose the maximum immediately intervenable feature if its Shapley value is greater than half of the maximum feature Shapley value.

The cases are shown in Table 6. The expert’s prescription and the suggestions of various algorithms are shown in Table 7. Overall we found that our method can be promising for automatically suggesting correct interventions. Our method’s suggested interventions agreed with the domain expert’s prescribed interventions 4 out of 6 times, but not in Cases 1 and 6. Additionally our method performed better than logistic regression and the highest overall XGBoost feature importance baselines.

In Case1, the expert believed the exact identity of the educational activity, a feature we did not include was the true cause of the wheel-spinning and the intervention would be to redesign that particular activity. While we did include the mini-game type of each activity, we did not include the unique identity of each activity in the model as it would result in too many features compared to the amount of data we had. Therefore one tradeoff of our method that needs to be made when there is limited data is using as many features as we can to catch all possible causes and using only the most important subset of the features to maintain model robustness. In Case 6, even though the prerequisite struc-

Features	Case1	Case2	Case3	Case 4	Case 5	Case 6
Mini-game (MG)	31	31.	611.	31	31.	546
Last Played (s) (LP)	34.	10.	6	10	12	25
First Time Seen? (F?)	F	F	F	F	F	T
Attempts Prereq1 (P1)	1	1.	1	7	12	\emptyset
Attempts Prereq2 (P2)	1.	1.	1.	1	1.	\emptyset
Attempts Prereq3 (P3)	1.	4.	1.	3	4.	\emptyset

Table 6: The 6 cases from the CWTL dataset used for qualitative evaluation of the methods.

	Expert	Ours	LR	FI
Case1	\emptyset	P3	P3	P3
Case2	P3	P3	P3	P3
Case3	MG	MG	MG	P3
Case4	P1	P1	P3	P3
Case5	P1	P1	P3	P3
Case6	P1	F?	F?	P3
Accuracy	-	4/6	2/6	1/6

Table 7: A comparison of our method and various baselines with the Expert’s prescription. Ours refers to the method described in this work, LR refers to the logistic regression baseline, and FI refers to the XGBoost overall feature importance baseline. MG refers to the “Mini-game” feature, F? refers to “First Time Seen?” feature, P1, P2, and P3, refer to “Attempts Prerequisite1”, “Prerequisite2” and “Prerequisite3” respectively and \emptyset refers to an expert prescription not in the list of what the model can suggest.

ture was created together with the domain expert, during the activity of prescribing interventions, the expert realized there may have been an incorrect dependency in the graph. Where under the original graph there were no prerequisites for this activity, under the new prerequisite graph this activity would have prerequisites. This case highlights the importance of having the correct curriculum graph.

In both the incorrect cases it would not have been feasible for our method to have obtained the correct answer, suggesting the ability of our method to identify correct interventions given correct inputs.

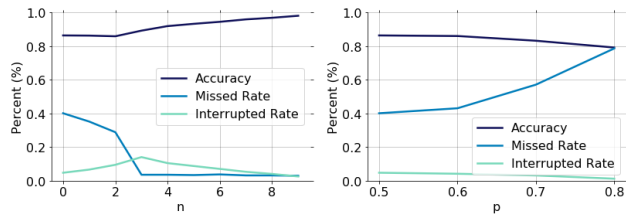
6. DISCUSSION

6.1 Possible Improvements With More Data

The program is currently running and data is being collected. As the amount of data increases and even more expressive function classes, such as neural networks, can be robustly trained, it is possible for the model to become more accurate. Additionally currently we have limited data, especially of the wheel-spinning class, therefore we do not include all possible helpful features, such as exact activity identity, to ensure model robustness. This omission can cause errors such as in Case 1. As more data becomes available this tradeoff between including features and model robustness becomes less important. More features can be included for more accurate intervention predictions.

Table 8: Parameters for Student Model 1

P(T) “easy”	P(T) “hard”	P(G) (A1,A2,A3)	P(G) (A4,A5)	P(S)	“hard” Skill
0.5	0.01	0.01	0.005	0	A2



(a) Sweeping n ($p = 0.5$) with Student Model 2 (b) Sweeping p ($n = 0$) with Student Model 2

Figure 4: Sweeping Hyperparameters

6.2 Setting Hyperparameters

As shown in both the results sections, our prediction at the 0th attempt of a student activity pair (before the student starts an activity) can be inaccurate. As we increase the number of attempts, n , before we intervene, we are able to increase accuracy as we by default do not intervene on students who need less than n attempts. However this increased accuracy comes at the expense of letting the students who will wheel spin spend time unproductively attempting the activity. This tradeoff may also not be feasible in environments where students may dropout before n attempts such as educational games played in a casual setting. We illustrate the miss rate decreasing and the accuracy increasing as we increase the number of attempts on which we make the prediction for Student Model 2 (Section 4.3) in Figure 4b. We fix the threshold probability of the wheel-spinning model output to make prediction (p) at 0.5.

Another key design choice touched upon is setting p , the threshold of the wheel-spinning model output for classifying wheel-spinning. To give a concrete example, changing the threshold from the default 0.5 to 0.7 would mean we need the wheel-spinning model to output a probability of 0.7 on a student-activity pair before we decide to suggest an intervention. Therefore at every attempt, we can trade off between correctly suggesting an intervention for a student-question pair and “interrupting” students by changing the certainty threshold. We examine this tradeoff using simulations following Student Model 2 (Section 4.3) at $n = 0$ and plot this in Figure 4b. As expected, as we increase the threshold, the missed rate increases as the interrupted rate decreases.

6.3 Limitation: Does Not Establish Causality

One limitation of this method is causal inferences cannot be made. To illustrate this we consider simulations following the simulation procedure of Student Model 1 (Section 4.2) under a new set of parameters given in Table 8. In this case we make A2 difficult instead of A1. As shown in Figure 2a, A2 only affects A4. Students who struggle due to unmastered prerequisite skills only struggle on A4. There will be very few students who, due to randomness, will struggle on A5. Therefore A4 will be positively correlated with wheel-spinning. However the design of A4 is not the direct cause of most students’ struggling where the true cause is the lack of mastery on A2. Looking into the Shapley values, A4 is chosen incorrectly as the highest valued feature for 11% of all true positive wheel-spinning cases. This can inaccurately lead to an assumption that A4 needs to be redesigned. While redesigning A4 could indeed reduce the number of students wheel-spinning on A4, if students master A2, they will not

struggle more on A4 than they would on A5. Therefore suggesting reviewing A2 instead of redesigning A4 as the most likely intervention candidate would be desired as reviewing A2 is often a much lower overhead intervention than redesigning A4. Coming up with solutions for this issue would be an interesting direction of future work.

7. CONCLUSIONS

In this work we propose a method to automatically suggest interventions for wheel-spinning students. To our knowledge this is one of the first investigations of both designing a wheel-spinning model to suggest immediately actionable interventions as well as using interpretable machine learning methods such as Shapley values in educational technology. We evaluate our method’s ability to suggest useful interventions by investigating the correctness of the suggested intervention in two different simulations and through a qualitative investigation comparing the interventions suggested by our method and the interventions prescribed by the expert. We found our method had high accuracy and was able to choose an accurate intervention for more than 80% of the time in the simulations before the students begin an activity. Additionally in our real world setting our suggestions mostly agreed with the expert prescription and the other cases were due to limitations of the model and errors made in the inputs to the model. Our results suggest our method can help inform interventions and improve educational systems to be more effective and engaging.

8. ACKNOWLEDGEMENTS

We would like to thank all our collaborators at War Child Holland and the Can’t Wait To Learn team. This research was supported by the NSF CAREER award, NSF big data award and the GFSD Fellowship.

9. REFERENCES

- [1] J. M. L. Andres and M. M. T. Rodrigo. The incidence and persistence of affective states while playing newton’s playground. In *7th IEEE international conference on humanoid, nanotechnology, information technology, communication and control, environment, and management*, 2014.
- [2] R. Baker, J. Walonoski, N. Heffernan, I. Roll, A. Corbett, and K. Koedinger. Why students engage in “gaming the system” behavior in interactive learning environments. *Journal of Interactive Learning Research*, 19(2).
- [3] R. S. Baker, A. T. Corbett, K. R. Koedinger, and A. Z. Wagner. Off-task behavior in the cognitive tutor classroom: when students’ game the system”. In *Proceedings of the SIGCHI conference on Human factors in computing systems*.
- [4] R. S. Baker, S. K. D’Mello, M. M. T. Rodrigo, and A. C. Graesser. Better to be frustrated than bored: The incidence, persistence, and impact of learners’ cognitive-affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, 68(4).
- [5] J. Beck and M. M. T. Rodrigo. Understanding wheel spinning in the context of affective factors. In

- [6] J. E. Beck and Y. Gong. Wheel-spinning: Students who fail to master a skill. In *International conference on artificial intelligence in education*.
- [7] F. Brown, A. Farag, F. Hussein, L. Miller, K. Radford, A. Abdullatif Abbadi, K. Neijenhuijs, H. Stubbe-Alberts, T. de Hoop, J. Turner, A. Jetten, and M. Jordans. Can't wait to learn: A quasi-experimental mixed-methods evaluation of a digital game-based learning programme for out of school children in sudan. *Under Review in the Journal of Development Effectiveness*.
- [8] G. Chanel, C. Rebetez, M. Bétrancourt, and T. Pun. Emotion assessment from physiological signals for adaptation of game difficulty. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(6):1052–1063, 2011.
- [9] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*.
- [10] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4).
- [11] Y. Gong and J. E. Beck. Towards detecting wheel-spinning: Future failure in mastery learning. In *Proceedings of the second (2015) ACM conference on learning@ scale*.
- [12] S. Kai, M. V. Almeda, R. S. Baker, C. Heffernan, and N. Heffernan. Decision tree modeling of wheel-spinning and productive persistence in skill builders. *JEDM| Journal of Educational Data Mining*, 10(1).
- [13] S. M. Lundberg, G. G. Erion, and S.-I. Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.
- [14] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*.
- [15] S. M. Lundberg, B. Nair, M. S. Vavilala, M. Horibe, M. J. Eisses, T. Adams, D. E. Liston, D. K.-W. Low, S.-F. Newman, J. Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering*, 2(10).
- [16] N. Matsuda, S. Chandrasekaran, and J. C. Stamper. How quickly can wheel spinning be detected? In *EDM*.
- [17] R. Narayanam and Y. Narahari. A shapley value-based approach to discover influential nodes in social networks. *IEEE Transactions on Automation Science and Engineering*, 8(1).
- [18] M. M. T. Rodrigo. Dynamics of student cognitive-affective transitions during a mathematics game. *Simulation & Gaming*, 42(1).
- [19] L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.
- [20] E. Štrumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3).
- [21] J. Turner, K. Taha, N. Ibrahim, K. I. Neijenhuijs, E. Hallak, K. Radford, H. Stubbe-Alberts, T. de Hoop, M. J. Jordans, and F. L. Brown. A mixed-methods evaluation of an innovative, digital game-based learning programme to improve educational outcomes of out-of-school children in lebanon. In *Submission to the Journal of Education in Emergencies*.
- [22] J. J. Vogel-Walcutt, L. Fiorella, T. Carper, and S. Schatz. The definition, assessment, and mitigation of state boredom within educational settings: A comprehensive review. *Educational Psychology Review*, 24(1).
- [23] H. Wan and J. B. Beck. Considering the influence of prerequisite performance on wheel spinning. *International Conference on Educational Data Mining*, 2015.
- [24] C. Zhang, Y. Huang, J. Wang, D. Lu, W. Fang, J. Stamper, S. Fancsali, K. Holstein, and V. Aleven. Early detection of wheel spinning: Comparison across tutors, models, features, and operationalizations. *International Conference on Educational Data Mining*, 2019.
- [25] Y.-J. Zhang, A.-D. Wang, and Y.-B. Da. Regional allocation of carbon emission quotas in china: Evidence from the shapley value method. *Energy Policy*, 74.

Exploring homophily in demographics and academic performance using spatial-temporal student networks

Quan Nguyen
University of Michigan
quanngu@umich.edu

Oleksandra Poquet
University of South Australia
sspoquet@gmail.com

Christopher Brooks
University of Michigan
brooksch@umich.edu

Warren Li
University of Michigan
liwarren@umich.edu

ABSTRACT

Network analysis in educational research has primarily relied on self-reported relationships or connections inferred from online learning environments, such as discussion forums. However, a large part of students' social connections through day-to-day on-campus encounters has remained underexplored. The paper examines spatial-temporal student networks using campus WiFi log data throughout a semester, and their relations to the student demographics and academic performance. A tie in the spatial-temporal network was inferred when two individuals connected to the same WiFi access point at the same time intervals at the 'beyond chance' frequency. Our findings revealed that students were more likely to co-locate with the individuals of similar gender, ethnic group identity, family income, and grades. Analysis of homophily over the semester showed that students of the same gender were more likely to co-locate as the semester progressed. However, co-location of the students similar on ethnic minority identity, family income, and grades remained consistent throughout the semester. Mixed-effect regression models demonstrated that features derived from spatial-temporal networks, such as degree, the grade of the most frequently co-located peer, and average grade of five most frequently co-located peers were positively associated with academic performance. This study offers a unique exploration of the potential use of WiFi log data in understanding of student relationships integral to the quality of college experience.

Keywords

Network analysis, homophily, spatial-temporal data, WiFi log data.

1. INTRODUCTION

With massification and globalization of higher education, students are exposed to individuals from a different nationality, ethnicity, gender, and socio-economic background. Universities have long been known as physical spaces where students form lifelong social connections, both for professional social capital and personal networks, such as friendship and marriage [1]. Therefore, understanding how social connections form and change in educational settings, as well as the impact student networks have on learning outcomes, can inform educators of unique ways to improve learners' experience [2].

Educational research offers a range of literatures focused on student networks in both face-to-face and blended or online settings. This paper explores homophily in demographics and academic performance using spatial-temporal student networks" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 194 - 201

settings [2-9]. Social scientists have conventionally derived student networks from self-report surveys [2, 10]. These surveys ask students to list who they are friends with or who they seek advice from [2]. The data can be collected multiple times to track the changes in network formation [10, 11]. Self-reported networks are a source for much of the extant evidence about student networks. However, such data collection is vulnerable to sampling biases (i.e. a low response rate, a sample from one class) where important network observations may be omitted. The timing of surveys may affect derived network features, and frequent surveying of learners can lead to survey fatigue and a lack of responses.

Instead of self-reports, the EDM and LAK communities have based their network studies on the log-data generated from online discussion forums [3, 4, 12-14]. Digital traces from online discussion enabled researchers to capture the structure, frequency, as well as the content of communication exchanges. Student networks constructed from online logs also have limitations. For instance, many online courses do not require that students use online forums. In face-to-face or blended learning settings, students are also less likely to use discussion forums. Therefore, student networks derived from online communication are limited in their generalizability, which remains a major challenge for researchers in this domain.

One underexplored data source for social network research in educational settings is location-based data. Social scientists have long argued that those in close physical proximity are more likely to form a social connection (McPherson, Smith-Lovin and Cook [1], p.430). More recently, relationship between geographic proximity and social ties have been corroborated by fine-grained geo-location-based analysis using mobile technologies. For example, the Copenhagen Networks Study [15] quantified the impact of physical proximity on student network structures using 500 GPS-enabled smart phones. Eagle, Pentland and Lazer [16] also used mobile technologies to compare the network based on physical proximity with the self-report social network and reported that 95% of the network friendships can be accurately inferred from sensor data. Although student location data from GPS and Bluetooth signals has shown to be informative, such methods are expensive to replicate and challenging to scale due to a high equipment cost.

This paper presents yet another source for location-based data to infer student networks. The paper reports on the study of student networks constructed from routinely collected WiFi logs. Such network data is created transparently to the learners as they connect to campus WiFi access points which are ubiquitous across physical campuses. Spatial-temporal ties between users can be inferred based on the overlap of time intervals in which learners connected to the same access point, suggesting a

reasonably close spatial co-location (room level). The study aims to understand the relationship between spatial-temporal ties and student characteristics across time, and predictive potential of the features derived from spatial temporal networks.

1.1 WiFi network data in education research

Wireless local area networks (WLANs) are ubiquitous in higher education as they provide on-campus Internet access to students, teachers, and staff. Despite extensive research using WiFi data, only a limited number of studies has explored their application for educational purposes [17-20]. A common example is the usage of WiFi data to visualize mobility patterns. For example, the iSpots project showed how people move around campus in real-time [17]. Hang, Pytlarz and Neville [20] combined WiFi logs with information about the buildings to extrapolate user preferences, and to predict user locations using graph embeddings. WiFi data has also been used in predictive modelling. Sarkar, Carpenter, Bader-El-Den and Knight [19] estimated the correlations between students' on-campus time based on WiFi logs and academic performance. Zhou, Ma, Zhang, SuiA, Pei and Moscibroda [18] utilized WLAN data to estimate students' punctuality for lectures to assess the lecture's engagement using mobile phone's interactive states at minute-scale granularity.

An application of WiFi data which has yet to be explored in areas such as EDM is the formation of social network among students on campus. In line with previous research on location-based networks [15, 16], social ties between WiFi users can be inferred from spatial and temporal co-occurrences (i.e. two users connected to the same WiFi access point during the same time window). Compared to surveys, discussion forum data, and proximity data collected through mobile devices (e.g. Bluetooth beacons), WiFi data provides a fine-grained alternative that records the dynamic changes in social interactions over a long period of time. Importantly, WiFi logs can capture physical social interactions and can scale at a relatively low cost. This paper presents initial steps towards exploring spatial and temporal information in the analysis related to student learning.

1.2 Research questions

Individuals are likely to share social connections with others similar to them, a phenomenon known as homophily [1, 21]. In educational settings, researchers have observed homophily based on gender [22], ethnicity [23], international/domestic country of origin [10], study major [24], socio-economic status [23], and academic performance [25, 26]. It might be expected that high-performing students seek friendship with other high-performing peers as part of their academic identity [27, 28], or that groups of high performing learners joined by lower performing learners will raise up those learners [29]. While there has been a large literature exploring the homophily effects in educational settings using traditional questionnaires or interactions in online learning environments, there remains a paucity of research that utilizes location-based data for such purposes. We hypothesize that students with similar traits are more likely to spend more time together on campus, i.e. in a spatial temporal co-occurrence from which a social connection can be inferred.

RQ1: How do demographic characteristics and grades affect the likelihood of spatial-temporal co-occurrence among students?

Second, we examine if spatial-temporal student network can capture social selection processes among students, also a phenomenon previously observed in social student networks.

'Social selection' refers to the choice to interact with others of similar status or value, and has been observed in various educational settings [27, 28]. With the increasing availability of digital data in education (i.e. LMS, online discussion forums), researchers are enabled to observe the dynamics of social selection processes with high temporal precision. In these regards, we are interested in understanding the temporal changes in the homophily effects of demographics and academic performance over time. For example, one might expect that at the beginning of the semester, students are more likely to form friendships based on similarity in demographic attributes as they have not acquired sufficient information about their peers' academic ability. One might also expect that as students approach the end of the semester, more social ties will be formed within similar performance groups. This leads us to our second research question:

RQ2: How does homophily based on demographic characteristics and academic performance change over time?

In addition to these questions, previous studies [7, 21, 26] have confirmed a positive relation between the degree of social integration/participation and academic performance. Motivated by this, we are interested in the predictive potential of 'peer effects' for grade performance using location-based network data. The relationship between that of a peer and one's characteristics has been studied for dormmates, as well as classmates, schoolmates, or children from the same neighborhood [29]. Administrative records of class co-enrolment have also been shown to capture this relationship in predictive models [24, 30]. Therefore, it would be reasonable to expect that spatial-temporal student networks can be useful for engineering features based on the peers a student is co-located with.

RQ3: How do network indices of spatial temporal networks relate to student performance?

2. METHODS

2.1 Datasets

Data in this study were collected from 3,915 students enrolled in five large STEM freshman courses at the University of Michigan, USA during the Fall semester of 2018. The selected courses include introductory physics, calculus, biology, chemistry, and psychology. Note that while these make up only a small fraction of all available offerings, they are considered to be foundational for a wide range of degree programs. That is, these courses serve as a gateway into the discipline, account for a significant portion of total credits registered, and are an integral part of one's academic career upon which we can leverage data collection to better understand the broad needs of incoming students and to improve instruction. The format of these courses is primarily didactic in nature, consisting of large lecture-style classes with hundreds of student enrollments. Content coverage is relatively stable between terms albeit with changing instructional teams, and the diverse student body, both in terms of demographics and measures of performance, was a key determinant in selecting these log data to represent students' first-year experience.

All data were de-identified. The dataset contained 91.7 million time-stamped entries recording log data between each device being connected to a particular WiFi access point. Each entry contained a unique user ID, a timestamp, a timestamp when a device was disconnected from a WiFi access point, a WiFi access point descriptor which (often) included a physical location such

as a building name and room number, and the device MAC address (Table 1).

Table 1. De-identified sample WiFi data

ID	Timestamp	Session End	Access point	MAC
A1234	2018-09-24 08:00:00	NA	TWC-1023	XYZ123
A1234	2018-09-24 08:02:00	NA	TWC-2013	XYZ123
B2314	2018-09-24 08:00:03	2018-09-24 08:00:55	BAHR-1210	XYZ125
C2153	2018-09-24 08:00:05	NA	CQTB-3734	XYZ121

The data were pre-processed by dropping all records generated by MAC addresses that were connected to access points within a single building, because they were likely to be stationary devices, such as computers at the libraries or lecture halls. Second, we computed a “connected time” feature for each user by subtracting two consecutive timestamps ($t_2 - t_1$). For example, the connected time for user A1234 to access point TWC-1023 was 2 minutes (Table 1). The connected time feature is important for the subsequent network modeling, which requires a co-located time between any two users. Since the connected time could be biased when a device became disconnected (i.e. students left the building), we removed all data entries which contained a session end’s timestamp. After the pre-processing, we retained 80.9 million records of 3,910 users, and these records were joined with the demographic information and final grades for a semester.

2.2 Analysis

2.2.1 Compute co-located time

To draw inferences about the network structure from WiFi data, we created an undirected weighted one-mode network (i.e. user-user). A tie’s weight equaled to the total amount of co-located time between two users. Figure 1 visualizes the temporal changes in WiFi access points of two users on a particular date from 08:00 to 20:00. These two users spent a large amount of time in the morning at a fixed WiFi access point, possibly attending a lecture. In the afternoon, these two users shared the same access points for 2 hours. After that, each user went on about their day to different areas on campus.

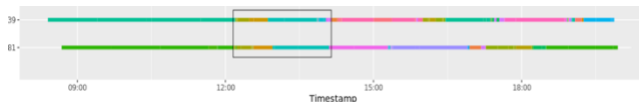


Figure 1: Temporal changes in WiFi access points of two users throughout a day. The boxed area indicates a two-hour period where these users shared the same access point

The co-located time between each pair of users was computed using the *roverlaps* package and stored in a 3910 x 3910 adjacency matrix.

2.2.2 Exponential random graph models (ERGMs)

RQ1 seeks to understand how demographic characteristics (e.g. gender, ethnicity, minority, under-representative, family income, parents’ educational level), and academic performance relate to the formation of ties amongst students. Specifically, we model if students from the same background or having the same academic performance were more likely to form a connection. We used Exponential Random Graph Model (ERGM) techniques which have been used to explore homophily in network formation in

educational data previously [9, 13, 14]. ERGM, also known as a p^* model, is a stochastic model that specifies the probability of the entire network as a function of its network properties [31].

$$P(Y = y) = \exp(\theta'g(y)) / k(\theta)$$

- Y is the network realization;
- y is the observed network;
- $g(y)$ is a vector of model statistics for network y ;
- θ is the vector of coefficients for those statistics, and
- $k(\theta)$ represents a normalizing factor, calculated as the sum of $\exp(\theta'g(y))$ over all possible networks.

This can be expressed as the conditional log-odds of a single tie between two actors i and j :

$$\text{logit}(Y_{ij} = 1|y_{ijc}) = \theta'\delta(y_{ij})$$

where θ is the coefficient and $\delta(y_{ij})$ is a change statistic.

To translate this into our context, ERGM was used to estimate the likelihood, expressed in conditional log-odds of two students being connected, given the similarity in their demographic characteristics and course grades. Model fit was examined with AIC and BIC (the lower the better model fit) and visual plots.

An important analytical decision was taken when weighted ties in our spatial temporal network were transformed into binary relations. To do so, we applied a filtering technique called dyadic thresholding. That is, a tie between two students would be kept when its weight was more than two standard deviations above the mean of all weights across all students. In other words, two users were considered to have a social connection when they spent a large proportion of their time on campus around each other.

To address RQ2, we applied a time window slicing technique to create separate ERGMs for a network that captured every month of activities from September to December. We then compared the changes in network homophily based on demographics and academic performance across four months.

Finally, for RQ3, network indices at the level of a node/student were incorporated in mixed-effect regression models. The models predicted grades as a function of demographics and network properties. To test the relationship between peer performance and predicted grade, we incorporated two features: 1) the average grade of the most frequently co-located peer, and 2) the average grade of five most frequently co-located peers into the model.

All the analyses were carried in R 3.6.2. ERGMs were fit using the *statnet* package [31, 32], mixed-effect regression models were run with the *lme4* package [33]. A simulated dataset and the code will be made available on Github.

(https://github.com/quan3010/EDM20_Nguyen).

3. RESULTS

3.1 Network description

The data for network construction was comprised of 80.9 million log events of 3,910 users over four months. From that, we derived a weighted, undirected network with over 6.54 million weighted ties. An average co-located time between two students was 0.98 hours, with a standard deviation of 12.37 hours. This weighted graph was converted into an unweighted graph network by setting a cut-off value equal to two standard deviations about the mean, i.e. 25.74 hours. Thus, in the modelled network two users shared a tie only if they spent at least 25.74 hours together over a four-month period. The final network had 3,910 users and a total of

18,704 ties. In such a network, the median number of ties was 8 with maximum of 63 ties. For 50% of the students the range of connections was from a minimum of 3 peers to a maximum of 14. The average co-located time between two users was 120 hours, with a minimum of 25.74 hours, median of 38 hours, and a maximum of 1397.62 hours.

Table 2. Frequency statistics of demographic and grades

Gender	N	Percentage
Male	1969	50.4%
Female	1941	49.6%
Ethnicity		
White	2207	56.4%
Asian	763	19.5%
Hispanic	337	8.6%
Mixed	214	5.5%
Not Indic	197	5.0%
Black	187	4.8%
Native American	5	0.1%
Minority status		
Non-minority	2365	60.5%
Minority	1346	34.4%
International	199	5.1%
Underep stats		
Non-Underrepresented Minority	3083	78.8%
Underrepresented Minority	628	16.1%
International	199	5.1%
Family income		
> \$200,000	1043	26.7%
\$150,000-\$199,999	355	9.1%
\$100,000-149,999	563	14.4%
\$75,000-\$99,999	243	6.2%
\$50,000-\$74,999	266	6.8%
\$25,000-\$49,999	366	9.4%
< \$25,000	217	5.6%
NA	847	21.7%
Grade_letter		
A-, A, A+	1295	33.1%
B-, B, B+	1671	42.7%
C-, C, C+	664	17.0%
Below D	140	3.6%
Withdraw	120	3.1%

¹ Household income is self-reported on admissions data.

NA	20	0.5%
----	----	------

Table 2 provides descriptive statistics for 3,910 students in this study. There was a rough balance in the number of female and male students. This is important since homophily can occur at random, for instance when a relative size of a subgroup is markedly different. White was the most frequent ethnicity, followed by Asian and Hispanic. A third of the sample identifies as an ethnic minority and 16.1% was categorized as under-represented minority. The family income distribute are right-skewed with over a quarter of students report household income of over \$200,000¹. Academic performance in this semester followed a bimodal distribution with of the majority of students performed at the A-range and B-range.

3.2 Homophily based on demographics and grades

Table 3 reports the results of three ERGM models. Model 1 serves as the baseline model, which accounts for the density of the network. The log-odds of a tie was -6.01 which translates to a probability of a tie exists equal to 0.24% (i.e. 18,704 ties divided by a total of 7.64 million possible ties).

In model 2, we added five nodal attributes, including gender, ethnicity, ethnic minority status, under-represented minority status, and family income, to explore homophily related to demographics. Our results showed that students from the same gender were more likely to form a tie than those with different gender, with the probability of a same-gender tie being 62%. Ethnicity and underrepresented minority status of the student did not have any statistically significant effect on the formation of network ties. This may be explained by the effect of the minority variable, which already accounted for ethnicity and unrepresented groups. Although a social connection was more likely to exist between students from the same minority group (i.e. non-minority, international, minority), the effect was marginal with a probability of only 54%. Family income also had a small effect on the formation of ties. The probability of ties to exist between two users with the same family income was 53%.

Table 3. Homophily effects of demographics and grades

	Model 1	Model 2	Model 3
ties	-6.010*** (0.007)	-6.375*** (0.016)	-6.444*** (0.017)
gender		0.479*** (0.015)	0.479*** (0.015)
ethnic		-0.006 (0.022)	-0.008 (0.022)
minority		0.157*** (0.021)	0.161*** (0.021)
underrep		0.004 (0.018)	-0.004 (0.018)
family_income		0.135*** (0.021)	0.132*** (0.021)
grade_letter			0.211*** (0.015)
AIC	262,286	261,101	260,913
BIC	262,300	261,184	261,010

Note: * $p < 0.1$; ** $p < 0.05$; *** $p < 0.01$

Coefficients calculated in log-odds, standard errors in brackets. Finally, in model 3, we added student's grades to examine homophily related to academic performance. The probability of a tie among same-grade students was 55%. To conclude, we observed a strong homophily network effect in gender, and marginal effects in minority identity, family income, and academic performance. Spatial-temporal networks also captured the commonly observed patterns of social homophily. This suggests that spatial-temporal networks reflected the social connections underpinning the co-location patterns.

The measures of homophily based on demographics have important implications to the understanding of diversity and inclusivity in higher education. The mere presence of structural diversity in student body (i.e. the proportional representation of groups of students from different backgrounds) does not guarantee the interactions between these diverse groups (Puritty et al., 2017). Homophily measures could serve as an indicator of how diverse and inclusive the social interactions between students are. A highly homophilous network could signal social segregation, and to some extent, inequality in student body as students are less likely to form a connection with peers who are demographically different than themselves. The use of WiFi data could support the design of physical spaces/educational activities that increase the likelihood of spatial co-occurrences between diverse groups of students.

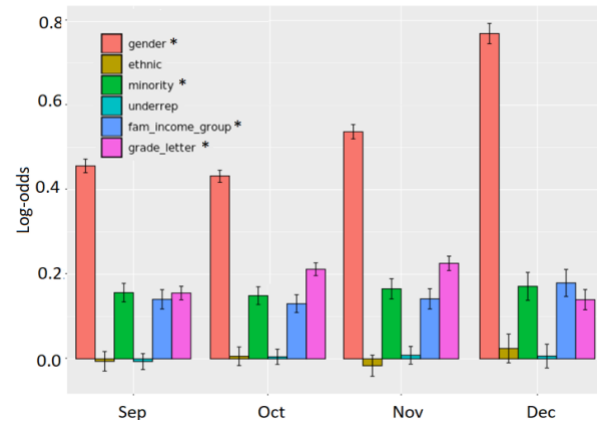
However, we are careful to draw inferences as to what the homophily represents. Our models do not control for types of building, or events that take place on campus. It is plausible that spatial temporal networks capture both the networks formed based on foci of activity (classes, living arrangements, cafeteria visits for students with similar schedules) as well as social ties. For instance, gender homophily could be explained by the majority of freshman students sharing their living space with same-gender peers in a residential building on campus. In this case, co-located time between roommates and dormmates would be the highest among freshmen. We did not find any evidence of homophily between different ethnicities per se. However, we observed homophily between different ethnic identities, such as ethnic minority (i.e. Black, Asian, Mixed, Hispanic), ethnic non-minority (i.e. White), and international (i.e. mostly Asian). In other words, there was evidence for inter-ethnic co-location within ethnic minorities.

As can be seen in the model, the addition of the terms decreased the AIC/BIC suggesting improved model fit. We did not manage to fit any of the conventional closure terms, such as popularity (e.g. geometrically weighted degree distribution) or transitivity (e.g. geometrically weighted edgewise shared partners), into the model. Visual examination of the goodness of fit suggested that the model was fit in predicting dyadic-level observations but was limited in reproducing the network structure. These results suggest that the model either requires to add control variables about the events/reasons for co-location (e.g. lectures, Thanksgiving breaks, exam periods), or that the networks need to be separated to have a more elaborate operationalization of co-location (e.g. residential building, libraries, classrooms).

3.3 Temporal changes in social networks

To examine the changes in the homophily over time, we ran the ERGM model with the same specification for a network capturing co-location in each month (Sep, Oct, Nov, Dec). The coefficients of each model were visualized in Figure 2.

Figure 2. Temporal changes in homophily effects of demographics and grades on network formation (* $p < 0.01$)



We can observe an increasing trend in homophily based on gender over time. The probability of a same-gender tie increased from 61% in September to 69% in December. There was a small increase in the homophily based on grade in October and November but it then decreased in December. The homophily effect of minority identity and family income remained constant over time.

One potential explanation for the increasing trend in gender-based homophily is that students started expanding their social circle with people in the same dorm hall/residential building, who are likely to have the same gender. This could also be explained by the participation in fraternity and sorority activities for freshman. As a result, we observed an increase in same-gender co-location over time as students formed new connections within a fraternity and sorority. Finally, previous studies [29, 30] also observed the intersectional nature of grade-based performance, i.e. high-performing boys are likely to form ties with high performing boys, and the same applies to girls. The consistent trend in performance-based homophily could be explained by the fact that this is the first semester and not only are students new to the institution, but university academic performance was generally not available until the end of the semester. Results suggest that it could be interesting to examine the temporal changes in performance-based homophily over a longer time period, especially in sophomore and senior students.

This finding has important implication to the research of social interactions between students. More often than not, social relations in educational research are collapsed under a static and dichotomous category (e.g. yes/no). In reality, the formation of social relations is a highly dynamic and time-variant process. For example, students could become closer with certain peers while more distant with others as time goes by. Students' social circle could be more elastic during their freshman year but gradually form a close-knit group as they approach their senior year. The networks inferred from WiFi data allow us to explore many questions about the evolution in social interactions between students over time, which could not previously be answered with self-report social network surveys.

3.4 Predicting academic performance

We applied mixed-effect regression models to control for the heterogeneity between courses (Table 4). Grade letters were converted into numeric format as per institutional guidelines, with a maximum value of 4.0. Our findings indicated that in the courses we studied, male students on average achieved 0.08 grade

points higher than female students. Compared to students with a family annual income over \$200,000, which accounted for a quarter of our dataset, students with a family income of \$75,000, \$50,000, and \$25,000 had 0.13, 0.30, and 0.43 grade points lower respectively. The effect of family income became marginal and non-statistically significant once it is above \$100,000. Students from an under-represented minority (i.e. Black, Hispanic, and Native American) also had on average 0.30 grade points lower than a non-underrepresented minority (i.e. Asian, Mixed, and White).

All three network indices had a positive and statistically significant relation with academic performance. On average, each additional tie increased a student's final course grade by 0.014 grade points. For each grade point increase in the most frequently co-located peer, the student's grade increased by 0.07 grade points. For each additional point increase in the average grade of the five most frequently co-located peers, a student's grade increased by 0.15 grade points. It is important to note that our results so not imply a causal relationship. The finding could be explained by a homophily effect (i.e. students co-located with similarly performed peers) or a roommate effect (i.e. performances of co-located peers influence a student's performance).

Table 4. Effects of demographics and network on grades

	Model 1	Model 2	Model 3	Model 4	Model 5
Male	0.055*	0.071**	0.075**	0.082**	0.080**
	(0.023)	(0.025)	(0.025)	(0.025)	(0.025)
Ethnicity (ref= White)					
Mixed	-0.036	-0.111	-0.065	-0.060	-0.070
	(0.135)	(0.177)	(0.175)	(0.174)	(0.174)
Asian	0.033	0.026	0.070	0.076	0.059
	(0.122)	(0.166)	(0.164)	(0.163)	(0.163)
Black	-0.358*	-0.304	-0.269	-0.254	-0.253
	(0.149)	(0.189)	(0.187)	(0.187)	(0.187)
Hispanic	-0.041	-0.049	-0.025	-0.012	-0.019
	(0.143)	(0.183)	(0.181)	(0.181)	(0.180)
Native Am	-0.295	-0.239	-0.141	-0.143	-0.129
	(0.357)	(0.374)	(0.369)	(0.368)	(0.367)
Ref = Non-minority					
Internatnl	0.239*	0.174	0.173	0.174	0.183
	(0.109)	(0.151)	(0.149)	(0.149)	(0.149)
Minority	0.043	0.067	0.028	0.020	0.033
	(0.125)	(0.168)	(0.166)	(0.165)	(0.165)
Ref = Non-underrepresented minority					
Underrep	-0.319***	-0.297***	-0.314***	-0.305***	-0.304***
	(0.081)	(0.090)	(0.089)	(0.089)	(0.089)
Family income (ref=above \$200,000)					
\$199,999		0.007	-0.007	-0.012	-0.010
		(0.042)	(0.042)	(0.042)	(0.042)
\$149,999		-0.082*	-0.091*	-0.085*	-0.082*
		(0.036)	(0.035)	(0.036)	(0.036)
\$99,999		-0.086	-0.081	-0.094*	-0.089
		(0.048)	(0.047)	(0.048)	(0.047)
\$74,999		-0.133**	-0.139**	-0.131**	-0.131**
		(0.048)	(0.047)	(0.047)	(0.047)

\$49,999		-0.280***	-0.309***	-0.303***	-0.298***
		(0.042)	(0.042)	(0.042)	(0.042)
\$25,000		-0.460***	-0.452***	-0.446***	-0.429***
		(0.053)	(0.053)	(0.053)	(0.053)
No. of ties			0.014***	0.014***	0.014***
			(0.002)	(0.002)	(0.002)
Closest peer				0.106***	0.072***
				(0.016)	(0.019)
5 close peers					0.150***
					(0.040)
Constant	3.114***	3.193***	3.059***	2.717***	2.345***
	(0.130)	(0.129)	(0.142)	(0.149)	(0.176)
Obs	4,422	3,554	3,554	3,500	3,500
AIC	9,872.995	7,945.483	7,871.099	7,733.857	7,726.519
BIC	9,956.122	8,068.999	8,000.792	7,869.388	7,868.211
Note:	*p<0.05; **p<0.01; ***p<0.001				

4. CONCLUSION

This paper explores the use of WiFi data of 3,910 students in Fall 2018 in understanding student physical on-campus connections. Specifically, we explore if spatial-temporal student networks reflect homophily based on demographics and academic performance expected in social networks. Network connections were inferred when two users exhibited a high level of co-located time (i.e. connecting to the same WiFi access point in the same time window). We found evidence of homophily with regards to gender, ethnic minority identity, family income, and academic performance. Gender-based homophily is particularly interesting, given that the composition of the student body has equal share of both genders and that this homophily increased significantly over time. This suggests that observed homophily is not baseline, but largely structural. That is, the organization of physical space, as well as curricular and extracurricular activities may create opportunities for gender-based homophily on campus. Exploring this further may be useful in understanding the effect of various institutional (e.g. gender-based meetups, structured study sessions, or mentoring workshops) and non-institutional (e.g. gender-based social activities, such as fraternity and sorority functions and enrollments) activities on the development of friend and support networks.

In addition, we found that the number of ties and the average performance of the most frequently co-located peer(s) were predictive of academic performance. This is in line with extant literature on self-reported peer effects [29], or the effects of peers observed from academic records [24]. Contextualizing this relationship and determining signals for specific causal activities is a clear next step.

From a theoretical perspective, our results confirmed homophily with regards to demographics and academic performance. At the same time, we extended the findings to capture the temporal changes in homophily within a semester. Our findings suggest that the tendency to form (co-located) connections may vary over time and more longitudinal studies are needed to understand the mechanism behind dynamic homophily.

From a methodological perspective, we demonstrated a novel application of spatial-temporal data in the study of student social networks, which have primarily relied on self-reports and log-data from discussion forums. This opens up a new venue to capture social interactions between students on campus on a large

scale and with fine-grained granularity. Importantly, this can be achieved without the need to collect additional data beyond what has been already collected by the university wireless networks. Location data inferred from WiFi access points can be considered as less invasive than using mobile phone's GPS or location-sensors to track users' location [15, 16]. Future research could combine self-report, discussion forum data, and location-based data to form a more holistic picture of student social networks and to triangulate findings from multiple data sources.

From a practical perspective, this study highlighted several factors that determine the formation of network among college students as well as their effect on academic performance. Such results may be useful to institutions in designing or evaluating location-based initiatives to promote gender, ethnicity, and culture diversity and inclusivity on campus, as well as to support ethnic minority, underrepresented minority in social integration during their time in college. There are opportunities to better understand the impacts of learning communities (e.g. themed residences for groups of students, such as Women in STEM communities), of co-curricular activities and their placement on campus (e.g. guest speakers or academic support groups), and even architectural planning (e.g. the relationship between dormitories and classrooms or libraries) through these methods.

4.1 Limitations

The data used does not capture the use of non-university run network (e.g. cellular networks), when students choose to go offline (e.g. intentionally by powering down their phone or due to low battery), or in spaces on campus without access to university network. There is also an inherent messiness which comes with the use of multiple or shared devices, the former of which is very common and increasing with the use of wearables. Network inference based on co-located time is further biased when students co-locate by random chance or by sharing common activities (i.e. attending lectures, going to the libraries, going to the gym) but do not interact with one another. Similarly, it is possible for students to be in completely different rooms yet connected to the same access point depending upon the wireless network and building topologies, introducing further noise to social network models. As a result, there might be hidden bias when using networks inferred from location data for predictive purposes. More sophisticated network inference techniques may be helpful in understanding this, such as weight/tie reshufflings or spatial/temporal simulations [34], and better cataloging of network endpoints (e.g. classroom, office, hallway) may be helpful in modeling social network relationships.

Finally, the modeling techniques used with the limited dataset chosen required significant computing power. More fine-grained temporal analyses (e.g. weekly or daily models), a longer time frame (e.g. a full academic year or throughout the students' academic career), and increased data (e.g. from more courses and non-freshman students) will only increase the need for computational power.

4.2 Concerns with the Use of Wi-Fi Data

WiFi data is highly sensitive data and the security of the collection, storage, and analysis of such data is of utmost importance. As is appropriate, we sought IRB oversight of our use of this data and worked with institutional data governance teams to ensure the data we received was appropriately stored, was de-identified, and was as minimal as possible to support our analyses.

At the same time, we feel it incumbent upon us to note that research access to such data is under threat by the potential misuse of educational location data for non-research purposes, which does not have to undergo IRB review. Specifically, some have begun to incorporate location data into formative evaluation of students. Location data is captured not only through WiFi, but also Bluetooth beacons and student mobile application software (which may be required), and has been used in an identifiable way to assign students grades based on location (attendance in class) [35]. While there are broad discussions of agency, privacy, paternalism, and ethics which the authors have positions on, the purpose of this section of the paper is to raise the importance such data has in understanding teaching and learning, and to encourage researchers in the field of Educational Data Mining (EDM) to voice opinions on the value of de-identified location data and its use in educational research.

5. ACKNOWLEDGMENTS

This work was funded under the Holistic Modeling of Education (HOME) project funded by the Michigan Institute for Data Science (MIDAS). We would also like to thank the UM's ITS team for their support with data access.

6. REFERENCES

- [1] McPherson, M., Smith-Lovin, L. and Cook, J. M. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27, 1, 415-444.
- [2] Grunspan, D. Z., Wiggins, B. L. and Goodreau, S. M. 2014. Understanding classrooms through social network analysis: A primer for social network analysis in education research. *CBE—Life Sciences Education*, 13, 2, 167-178.
- [3] Xu, Y., Gitinabard, N., Lynch, C. and Barnes, T. 2019. What You Say is Relevant to How You Make Friends: Measuring the Effect of Content on Social Connection. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)*. Montreal, Canada
- [4] Xu, Y., Lynch, C. F. and Barnes, T. 2018. How Many Friends Can You Make in a Week?: Evolving Social Relationships in MOOCs over Time. In *Proceedings of the 11th International Conference on Educational Data Mining (EDM 2018)*. Buffalo, New York
- [5] Wise, A. F. and Cui, Y. 2018. Learning communities in the crowd: Characteristics of content related interactions and social relationships in MOOC discussion forums. *Computers & Education*, 122, 221-242.
- [6] Fincham, E., Gašević, D. and Pardo, A. 2018. From Social Ties to Network Processes: Do Tie Definitions Matter? *Journal of Learning Analytics*, 5, 2, 9-28-29-28.
- [7] Dowell, N. M., Skrypnik, O., Joksimovic, S., Graesser, A. C., Dawson, S., Gašević, D., Hennis, T. A., de Vries, P. and Kovanovic, V. 2015. Modeling Learners' Social Centrality and Performance through Language and Discourse. In *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*. Madrid, Spain
- [8] Rabbany, R., Elatia, S., Takaffoli, M. and Zaiane, O. R. 2014. Collaborative Learning of Students in Online Discussion Forums: A Social Network Analysis Perspective. Springer International Publishing.

- [9] Zhu, M., Bergner, Y., Zhang, Y., Baker, R., Wang, Y. and Paquette, L. 2016. Longitudinal engagement, performance, and social connectivity: a MOOC case study using exponential random graph models. In *Proceedings of the 6th International Conference on Learning Analytics & Knowledge*. 223–230. Edinburgh, United Kingdom
- [10] Rienties, B. and Tempelaar, D. 2018. Turning Groups Inside Out: A Social Network Perspective. *Journal of the Learning Sciences*, 27, 4, 550-579.
- [11] Chen, B., Chang, Y.-H., Ouyang, F. and Zhou, W. 2018. Fostering student engagement in online discussion through social learning analytics. *The Internet and Higher Education*, 37, 21-30.
- [12] Poquet, O. and Dawson, S. 2016. Untangling MOOC learner networks. In *Proceedings of the 6th International Conference on Learning Analytics & Knowledge*. 208-212.
- [13] Poquet, O., Dowell, N., Brooks, C. and Dawson, S. 2018. Are MOOC forums changing? In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*. 340-349.
- [14] Joksimović, S., Manataki, A., Gašević, D., Dawson, S., Kovanović, V. and De Kereki, I. F. 2016. Translating network position into performance: importance of centrality in different network configurations. In *Proceedings of the 6th International Conference on Learning Analytics & Knowledge*. 314-323.
- [15] Stopczynski, A., Pentland, A. S. and Lehmann, S. 2018. How Physical Proximity Shapes Complex Social Networks. *Scientific Reports*, 8, 1, 17722.
- [16] Eagle, N., Pentland, A. S. and Lazer, D. 2009. Inferring friendship network structure by using mobile phone data. *Proceedings of the national academy of sciences*, 106, 36, 15274-15278.
- [17] Sevtsuk, A. 2009. Mapping the MIT campus in real time using WiFi. IGI Global.
- [18] Zhou, M., Ma, M., Zhang, Y., Sui, A., K., Pei, D. and Moscibroda, T. 2016. EDUM: classroom education measurements via large-scale WiFi networks. In *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*. 316-327.
- [19] Sarkar, S., Carpenter, B., Bader-El-Den, M. and Knight, A. 2016. Where students go and how they do: Wi-Fi location data versus academic performance. In *Proceedings of the 9th International Conference on Human System Interactions (HSI)*. 45-51.
- [20] Hang, M., Pytlarz, I. and Neville, J. 2018. Exploring student check-in behavior for improved point-of-interest prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 321-330.
- [21] Smirnov, I. and Thurner, S. 2017. Formation of homophily in academic performance: Students change their friends rather than performance. *PloS one*, 12, 8.
- [22] Stehlé, J., Charbonnier, F., Picard, T., Cattuto, C. and Barrat, A. 2013. Gender homophily from spatial behavior in a primary school: A sociometric study. *Social Networks*, 35, 4, 604-613.
- [23] Smith, J. A., McPherson, M. and Smith-Lovin, L. 2014. Social distance in the United States: Sex, race, religion, age, and education homophily among confidants, 1985 to 2004. *American Sociological Review*, 79, 3, 432-456.
- [24] Gardner, J. and Brooks, C. 2018. Coenrollment networks and their relationship to grades in undergraduate education. In *Proceedings of the 8th international conference on learning analytics and knowledge*. 295-304.
- [25] Gitinabard, N., Khoshnevisan, F., Lynch, C. F. and Wang, E. Y. 2018. Your actions or your associates? Predicting certification and dropout in MOOCs with behavioral and social features. In *Proceedings of the 11th International Conference on Educational Data Mining (EDM 2018)*. Buffalo, New York
- [26] Fire, M., Katz, G., Elovici, Y., Shapira, B. and Rokach, L. 2012. Predicting Student Exam's Scores by Analyzing Social Network Data. In *Proceedings of the International Conference on Active Media Technology* 584-595. Berlin, Heidelberg
- [27] Vaquero, L. M. and Cebrian, M. 2013. The rich club phenomenon in the classroom. *Scientific Reports*, 3, 1, 1174.
- [28] Kretschmer, D., Leszczensky, L. and Pink, S. 2018. Selection and influence processes in academic achievement—More pronounced for girls? *Social Networks*, 52, 251-260.
- [29] Sacerdote, B. 2001. Peer effects with random assignment: Results for Dartmouth roommates. *The Quarterly journal of economics*, 116, 2, 681-704.
- [30] Gašević, D., Zouaq, A. and Janzen, R. 2013. “Choose your classmates, your GPA is at stake!” The association of cross-class social ties and academic performance. *American Behavioral Scientist*, 57, 10, 1460-1479.
- [31] Hunter, D. R., Handcock, M. S., Butts, C. T., Goodreau, S. M. and Morris, M. 2008. ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of statistical software*, 24, 3, nihpa54860.
- [32] Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M. and Morris, M. 2008. statnet: Software tools for the representation, visualization, analysis and simulation of network data. *Journal of statistical software*, 24, 1, 1548.
- [33] Bates, D., Mächler, M., Bolker, B. and Walker, S. 2015. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67, 1, 1-48.
- [34] Poquet, S., Tupikina, L. and Santolini, M. 2019. Are forum networks social networks? A methodological perspective. In *Proceedings of the 10th International Conference of Learning Analytics & Knowledge (LAK20)*. in press. Frankfurt, Germany
- [35] Harwell, D. 2019. *The Washington Post*. Retrieved from <https://www.washingtonpost.com/technology/2019/12/24/colleges-are-turning-students-phones-into-surveillance-machines-tracking-locations-hundreds-thousands/>

The effect of teachers reassigning students to new Cognitive Tutor sections

Adam C Sales
University of Texas College of Education
Austin, TX, USA
asales@utexas.edu

John F Pane
RAND Corporation
Pittsburgh, PA, USA
jpane@rand.org

ABSTRACT

The design of the Cognitive Tutor Algebra I (CTA1) intelligent tutoring system assumes that students work through sections of material following a pre-specified order, and only move on from one section to the next after mastering the first section's skills. However, the software gives teachers the flexibility to override that structure, by reassigning students to different sections of the curriculum. Which students get reassigned? Does reassignment hurt student learning? Does it help? This paper used data from the treatment arm of a large effectiveness study of the CTA1 curriculum to estimate the effects of reassignment on students' scores on an Algebra I posttest. Since reassignment is not randomized, we used a multilevel propensity score matching design, along with assessments of sensitivity to bias from unmeasured confounding, to estimate the effects of reassignment. We found that reassignment reduces posttest scores by roughly 0.2 standard deviations—about the same as the overall CTA1 treatment effect—that unmeasured confounding is unlikely to completely explain this observed effect, and that the effect of reassignment may vary widely between classrooms.

1. INTRODUCTION

Two closely related pillars of intelligent tutoring systems are sequencing and mastery learning. It has long been obvious that the sequence in which students learn different topics is an important component of a curriculum, due to prerequisites—for instance, students must master arithmetic in order to learn how to solve algebraic equations. A related example is scaffolding, in which learners gradually achieve independence over a sequence of problems; scaffolding “consists essentially of the adult ‘controlling’ those elements of the task that are initially beyond the learner’s capacity, thus permitting him to concentrate upon and complete only those elements that are within his range of competence” [33]. However, measuring the effects of sequencing [21] [9] and determining prerequisites or optimal sequences [29] [31] [17] remains an active area of research.

By “mastery learning,” we mean the idea that students should “progress through topics as they master them,” [22] as opposed to at a fixed pace. This typically results in students within the same classroom working on different parts of a curriculum at the same time.

The Cognitive Tutor Algebra I (CTA1) system [8] includes both features. A particular Algebra I curriculum is programmed into the software, so that students, if left alone, will encounter topics in a specific, intentional sequence. Mastery learning governs how they progress from one section to the next: an underlying knowledge tracing model estimates the probability students have mastered a set of pre-defined skills as they work through problems that incorporate those skills. Students ideally progress from one section to the next only after demonstrating mastery on the previous section's skills.

Mastery learning does not always proceed this way in the CTA1 software. After a student has worked a certain, pre-specified number of problems in a particular section, he or she is automatically promoted to the next section, even if he or she has not mastered its skills [28]. Teachers can also reassign students working on one section to work on an entirely different section. If a teacher reassigns a student to a section other than the next one in the sequence, reassignment violates the intended sequencing as well as mastery learning.

There are a number of reasons teachers may want to meddle in the automatic progress of students through a curriculum [16]. If a teacher observes an advanced student spending time on basic skills, the teacher may move the student to more advanced sections. If certain skills will be on a standardized test, and a teacher wants all students to have had exposure to those skills before the test, the teacher may reassign all of his or her students to work on a section covering those skills. If a teacher notices a student falling behind his or her peers in the classroom, the teacher may choose to reassign the student to the section that the rest of the class is working on, even if the student has not demonstrated mastery on prerequisite skills (at least, within the tutor). If a teacher disagrees with the method a certain CTA1 section employs in teaching an Algebra topic, the teacher may reassign students out of that section, perhaps to the next unit or section in the curriculum.

It is unclear whether reassignment benefits students. On the

Adam Sales and John Pane "The effect of teachers reassigning students to new Cognitive Tutor sections" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 202 - 211

one hand, it violates the design principles of the software. On the other hand, it allows teachers flexibility to teach the material as they see fit, and use the tutor to meet the particular needs of their classrooms.

This paper uses data from a large randomized trial of the CTA1 curriculum to estimate the effect of reassignment. Unfortunately for our purposes, reassignment itself was not randomized—the study was designed to estimate CTA1’s effectiveness, so access to the tutor was randomized instead. Still, log data from study participants includes data on how often each student was reassigned from one section to another, and posttests measure their algebra skills at the end of the study. For those reasons, this data provides a rare opportunity to measure the effect of reassignment, and, by extension, the (joint) importance of topic sequencing and mastery learning.

The following section gives background on the effectiveness trial and describes the data we will use for the study. Section 3 describes propensity score matching, the method we employ. Section 4 describes the propensity score models, which in turn describe characteristics of students who are reassigned. Section 5 describes the matching algorithm and covariate balance. Section 6 gives our main results on the effects of reassignment, including sensitivity analysis to confounding from unmeasured covariates and between-classroom effect heterogeneity. Section 7 concludes.

2. DATA: THE RAND CTA1 EFFECTIVENESS STUDY

In the years 2007–2010, the RAND Corporation conducted a randomized study to test the effectiveness of the CTA1 curriculum relative to business as usual. The study tested CTA1 under authentic, natural conditions—that is, oversight and support of CTA1’s use was the same as it would have been outside of an RCT. The study population consisted of over 25,000 students in 73 high schools and 74 middle schools located in 52 diverse school districts in seven states. Students in Algebra I classrooms in participating schools took an algebra I pretest and a posttest, both from the CTB/McGraw-Hill Acuity series. The pretest was the Algebra Readiness Exam, a 40-item multiple-choice exam testing students’ algebra I prerequisite skills. The posttest was the Algebra Proficiency Exam, a 32-item multiple-choice exam testing algebra I skills including solving equations for an unknown, graphing linear and quadratic functions, calculating complex algebraic expressions and other skills. Data from both exams were scored with a three-parameter item response theory (IRT) model.

Results [19] were reported separately for middle and high schools, in the first and second years of implementation. In the first year, estimated effects were close to zero in middle schools and slightly negative in high schools, with confidence intervals including negative, null, and positive effects in both cases. In the second year, estimated effects were positive—roughly one fifth of a standard deviation—in both middle and high schools, and were statistically significant in high schools. In the high school sample, the difference between the effects in the first and second years was statistically significant as well.

Table 1: The number and percent of students in each study year of the dataset who were never reassigned, or reassigned once, twice, three times, or four or more times

year		# Reassignments				
		0	1	2	3	4+
1	n	1621	552	133	43	34
	%	68	23	6	2	1
2	n	1056	297	193	95	194
	%	58	16	11	5	11

As part of the study, RAND collected basic demographic data from students, including gender, race/ethnicity, prior standardized test scores, and special education, free or reduced-price lunch, and English language learner status.

Carnegie Learning collected computer log data from most users in the treatment arm of the study. At the problem level, this dataset records which problems students attempted, along with timestamps and the numbers of hints and errors for each attempted problem. The dataset also contains data on which sections of CTA1 students attempted, and the result: whether the student mastered the section, was promoted automatically without mastery, was reassigned by the teacher to a new section, or stopped using the tutor altogether midway through the section.

The current study analyzes data from the high school treatment group only, assessing the effect of teachers reassigning students from one CTA1 section to another. Since students in the control arm of the study did not have access to the tutor, section reassignment is not relevant for them. We focus on high school, as opposed to middle school, since the characteristics of Algebra I students tend to differ between the two levels: 8th-grade students only take Algebra I if they are sufficiently advanced, whereas most 9th grade students (who have not taken it already) take Algebra I regardless. Thus, the high school sample was not only larger but also more broadly representative than middle school sample.

Unfortunately, log data was not available for every student in the treatment arm of the study, primarily for two reasons: some students in CTA1 schools nevertheless did not use the tutor, and some students used the tutor but their log data was irretrievable or could not be reliably linked to posttest scores and covariates. This study omitted schools in which data was missing for over 20% of students in either year, leaving 18 schools. Among the students at these schools, we omitted 164 who had no log data, and 242 who worked—but did not complete—only one section or who had no section completion data for some other reason. A total of 4,218 students in 282 classrooms remained in the analysis sample, roughly 70% of the full treatment group.

Table 1 shows the number of included students in each year of the experiment who were reassigned zero, one, two, three, or four or more times. Since the sample size decreases quickly with the number of reassignments, and for the sake of simplicity, we chose to dichotomize reassignment, esti-

inating the effect of being reassigned at least once versus never.

3. STATISTICAL APPROACH

For subjects $i = 1, \dots, N$ in the treatment arm of the CTA1 trial, let Y_i denote subject i 's posttest score, and let $Z_i \in \{0, 1\}$ indicate whether i was ever reassigned. Following [18] and [25], let y_i^0 and y_i^1 denote i 's posttest score were $Z_i = 0$ or 1—i.e., had i not been reassigned, or had i been reassigned, perhaps counterfactually—and let $\tau_i = y_i^1 - y_i^0$ be the effect of reassignment on i 's posttest score. Since y_i^1 and y_i^0 are never simultaneously observed, τ_i is unidentified; however, weighted average treatment effects of the form $\tau^w = \sum_i w_i \tau_i$, with $w_i \geq 0$ and $\sum_i w_i = 1$ may be identified under the right causal assumptions. For instance, had Z been randomized, the average treatment effect, τ^w with $w_i = 1/N$, could be estimated without bias by the difference in the mean of Y between subjects with $Z = 1$ and with $Z = 0$. Of course, reassignment Z was not random, so identifying average treatment effects requires some combination of control for observed covariates and assumptions about unobserved covariates.

Let \mathbf{x}_i denote a vector of covariates for subject i . These include pretest scores, special education, gifted, and English language learner (ELL) status, race/ethnicity (white, black, Latinx¹), received free or reduced-price lunch (FRL). Let $Class_i$ be i 's classroom; since reassignment occurred within classrooms, $Class$ is a covariate as well. If reassignment were randomly assigned, the (theoretical) distribution of \mathbf{x} and $Class$ would be equal between reassigned and not-reassigned students— \mathbf{x} and $Class$ would be balanced. Our strategy will be to construct a randomization scheme in which \mathbf{x} , and, to the extent possible, $Class$ are balanced, and conduct inference under that randomization scheme.

Specifically, we use propensity score matching [23] [27]. The propensity score for subject i , $e_i(\mathbf{x}_i, Class_i) = Pr(Z_i = 1 | \mathbf{x}_i, Class_i)$ is the probability of i being reassigned conditional on covariates \mathbf{x} and classroom. [24] showed that under two conditions, described below, estimates of the average treatment effect conditional on $e(\mathbf{x}, Class)$ are unbiased. To estimate effects, we first estimate propensity scores (Section 4), then identify groups of reassigned and not-reassigned students with similar estimated propensity scores—a “match”—and verify that covariates are sufficiently balanced within the matched sample (Section 5), and, finally, estimate effects within the matched sample 6.

The first condition for propensity score matching is that there is some randomness in the treatment assignment:

$$0 < e_i(\mathbf{x}_i, Class_i) < 1 \text{ for all } i. \quad (1)$$

When (1) fails for a subset of the analysis sample, common practice is to drop that subset and estimate average effects for the remainder of the analysis sample, i.e. the subset for which (1) holds; this subset is referred to as the “region of

¹For the sake of parsimony, these categories were collapsed from a larger set in the original dataset, so that 8 American Indian/Alaskan Native students were categorized as Latinx, 23 Asian/Pacific Islander students and 118 students with missing data were categorized as white, and 22 Other/Multiracial students were categorized as black.

common support” [4] [30]. In this study, including *Class* among the covariates leads to violations of (1). Of the 282 classrooms over the two years of the study, 95 contained no reassigned students, and in 52 classrooms every student was reassigned at least once. In this subset of the data, including 44% of students, $Pr(Z = 1 | Class) = 0$ or 1. Our solution is to drop classrooms in which no one or everyone was reassigned, and only estimate effects for students in classrooms with some reassignment variance, a student-level analysis.

We attempted a parallel classroom-level analysis, in which we matched classrooms in which all students were reassigned to classrooms in which no one was. However, we were unable to construct a match with adequate covariate balance (there were few no-reassigned classrooms with similar mean pretest scores to the all-reassigned classrooms that were of similar sizes). For that reason, we dropped the classroom-level analysis.

The second condition for propensity score matching is that there are no unmeasured confounders:

$$(y^1, y^0) \perp\!\!\!\perp Z | \mathbf{x}, Class \quad (2)$$

Assumption (2) is well known as the Achilles heel of causal inference outside of RCTs. (2) is untestable; its believability depends on what is understood about the process that underlies treatment assignment Z , and what covariates are available for control. In our case, reassignment is poorly understood, and appears highly idiosyncratic [16]. Fortunately, our study includes a pretest measure, and observational studies controlling for pretest scores tend to perform well, and replicate experimental estimates [6] [7]. Section 6.1 discusses a sensitivity analysis that relaxes 2 and assumes reasonable levels of unmeasured confounding.

Our attitude towards propensity score matching is agnostic. If the propensity score models in the following section were approximately correct, and yielded good estimates of the true propensity scores, then the theory underlying propensity score adjustment holds. If not, the process of propensity score matching may still result in a set of matched reassigned and not reassigned students that, on average, resemble each other on all measured covariates. In other words, the (mis)estimated propensity scores \hat{e} may still be approximate “balancing” scores, satisfying

$$\mathbf{x} \perp\!\!\!\perp Z | \hat{e}. \quad (3)$$

Causal inference based on comparisons within these matched sets will still be plausible; indeed, [24] showed that in order to estimate average treatment effects, it is sufficient to condition on a balancing score, rather than the propensity score itself.

Following that logic, we choose propensity score models, and matching schemes based on the fitted models, in order to satisfy (3). Since posttest scores play no role in propensity score estimation and matching, the process may be iterative without affecting the objectivity of the final causal estimate. That is, we may try a series of candidate propensity score models and matches, and choose the one that results in the best covariate balance. Only then do posttests enter the picture, so that we may estimate effects.

All data analysis was done in R [20] using the `tidyverse` suite of packages [32] for data manipulation, plotting, and other tasks. This document was produced dynamically with `knitr` [34]. Source code is available at www.github.com/adamSales/cpEffect.

4. PROPENSITY SCORES: WHO GETS REASSIGNED?

We use multilevel logistic regression [10] to estimate student level propensity scores. The multilevel regression accounts for the nesting of students within classrooms, classrooms within teachers, and teachers within schools. In constructing the model, we give special consideration to the role of pretest scores, a proxy for student mathematical ability at the beginning of the school year, in predicting reassignment. First, we decompose pretest scores into student- and classroom-level components. If w_i is student i 's pretest score, let $w_i = \bar{w}_{j[i]} + \tilde{w}_i$, where $\bar{w}_{j[i]}$ is the average pretest score in i 's classroom $j[i]$, and \tilde{w}_i is the difference between i 's pretest score and the classroom mean. This decomposition was motivated by the possibility that reassignment patterns may differ between high- and low-achieving classrooms, and that a teacher's decision to reassign a student depends on the student's ability relative to the classroom than his or her absolute ability. Second, we modeled the effect of \tilde{w} on Z as linear in the logit scale, but allowed the slope to vary by classroom. This was motivated by the possibility that some teachers use reassignment to help struggling students catch up to their peers, so lower \tilde{w} would predict Z , and other teachers use it to help high-achievers skip sections related to basic skills, so higher \tilde{w} would predict Z . We also considered models incorporating non-linear effects of \tilde{w} , via natural splines [14] but found no evidence that the non-linearity improved the model fit. We fit the model using the `lme4` package in R [1].

All in all, the propensity score model was:

$$\begin{aligned} \text{logit} \{Pr(Z_i = 1 | \mathbf{x}_i, \text{Class}_i = j)\} = & \\ & \beta_{0state[i]} + \beta_1 \tilde{w}_i + \beta_2 \bar{w}_{j[i]} + \\ & \beta_3 \text{Black}_i + \beta_4 \text{Latinx}_i + \beta_5 \text{Male}_i + \\ & \beta_6 \text{Freshman}_i + \beta_7 \text{SpEd}_i + \beta_8 \text{gifted}_i + \\ & \beta_9 \text{ESL}_i + \beta_{10} \text{FRL}_i + \beta_{11} \text{FRLmis}_i + \beta_{12} \text{year}_i + \\ & \gamma_{j[i]} \tilde{w}_i + \epsilon_{j[i]}^{Cls} + \epsilon_{k[i]}^{Teach} + \epsilon_{l[i]}^{Schl} \end{aligned} \quad (4)$$

where $\text{logit}(x) = \log(x/(1-x))$ is the logit function, $\beta_{0state[i]}$ is a (fixed) intercept for each state in the sample, FRLmis_i is an indicator for missing data in *FRL* (which was mode-imputed), and $\text{year}_i = 1, 2$ is the study year for subject i .

Finally, $\gamma_{j[i]}$, $\epsilon_{j[i]}^{Cls}$, $\epsilon_{k[i]}^{Teach}$, and $\epsilon_{l[i]}^{Schl}$ are random effects. The subscripts j , k and l refer to classroom, teacher, and school, respectively; the $[i]$ refers to student, so that $j[i]$ is i 's classroom, $k[i]$ is i 's teacher, and $l[i]$ is i 's school.

$\gamma_{j[i]}$ is a random slope for \tilde{w}_i , varying at the classroom level. This is essentially an interaction term, allowing the slope for (classroom centered) pretest scores to vary from one classroom to the next. However, unlike standard regression interactions, random slopes are modeled as being drawn from a normal distribution, with a standard deviation estimated from the data. This is a form of regularization, shrinking

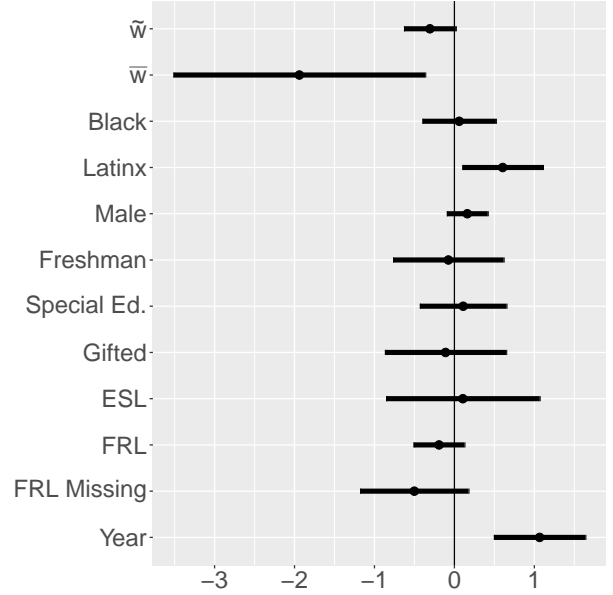


Figure 1: Estimated coefficients and 95% confidence intervals for student and class-level covariates from model (4).

the classroom-level slopes towards a common value, and allowing stable estimation even with very few observations from each classroom [10] [26]. The set of random slopes γ_j has a mean of zero—the average slope across classrooms is the fixed intercept β_1 . Therefore, the slope for pretest in classroom j is $\beta_1 + \gamma_j$.

ϵ_j^{Cls} , ϵ_k^{Teach} , and ϵ_l^{Schl} are random intercepts for classroom, teacher, and school. These were also modeled as normal with a mean of zero and a standard deviation estimated from the data. Including them in the regression accounts the fact that two students in the same classroom or with the same teacher or in the same school may be more likely to have the same Z —either both be reassigned or neither—than two students in different classrooms, with different teachers, or in different schools.

Figure 1 gives estimated coefficients and 95% confidence intervals for the propensity score model (4). Reassignment was much more prevalent in the second year of implementation than in the first, and classrooms with low average pretest scores reassigned students more often—though the magnitude of this trend is hard to determine, ranging from moderate to very large (the coefficients for \tilde{w} and \bar{w} were scaled by the standard deviations of these variables in the data). Latinx students were reassigned more often than their White classmates.

Students with lower pretest scores were reassigned more frequently than their classmates with higher scores. However, this may vary by classroom. On average, classroom-specific β_{1j} was approximately -0.31 standard deviations, but the 95% confidence interval for the mean includes slightly positive values as well. The standard deviation of β_{1j} , varying by classroom, was estimated as 0.83, suggesting that in some

classrooms the slope on \tilde{w} was moderately positive, and in others it was negative. However, the model was not able to estimate the variance of β_{1j} precisely; the p-value testing the null hypothesis of zero variance was 0.07.² When model (4) was modified so that β_1 was not allowed to vary by classroom, it was estimated as -0.32 ± 0.27 .

5. MATCHING AND COVARIATE BALANCE

We construct a student-level match based on propensity scores on the log-odds scale, i.e. $\log(\hat{e}/(1 - \hat{e}))$. Instead of a pair-matching design, which would necessitate discarding non-reassigned students who would make good matched comparisons, we use a restricted full match design [11]. In this design, the numbers of reassigned and not-reassigned students in each matched set is allowed to vary, so that in some cases several reassigned students may be matched with a single non-reassigned student, and vice-versa. We use the R package `optmatch` [13] to choose the matched sets optimally. The `fullmatch()` routine takes a matrix of discrepancies (e.g. differences in propensity scores) between treatment and control subjects, and arranges them into matched sets so that the sum of absolute discrepancies between matched subjects is minimized.

As described at the end of Section 3, the post-test scores played no role in this process. Hence, we were able to iteratively match students, check covariate balance, modify the propensity score model and/or the matching routine if necessary, and repeat until adequate balance was achieved. Here we present the final match; a record of attempts is available on the first author's github site.

The initial full match based on the log-odds propensity scores yielded decent covariate balance. However, pretest scores were slightly unbalanced, and since we consider pretest to be the most important covariate, we decided to match on the Mahalanobis distances between reassigned and not reassigned students combining propensity scores and pretest scores. Additionally, as displayed in Figure 2, the distributions of propensity scores among reassigned and not-reassigned students do not entirely overlap. Although this is at least partially due to overfitting the propensity score model (4), matching students with highly discrepant propensity scores may hinder the believability of the result. Hence, in our final match we imposed a caliper of 0.3 pooled standard deviations of the Mahalanobis distances. This prevented students with very different pretest scores or propensity scores to be matched. On the other hand, matches were unavailable for 25% of the students in the sample (21% of reassigned students and 28% of not-reassigned students). Propensity scores for these students are colored red in Figure 2. Our effect estimates pertain only to the remaining 75% of students—all in all, 1480 students, 604 reassigned and 876 not reassigned.

Covariate balance after matching was excellent. Figure 3 and Table 2 give covariate balance (standardized differences) before and after matching. They were produced with the RI-

²This hypothesis was tested with a likelihood ratio χ^2 test comparing (4) to a model in which β_1 did not vary by classroom.

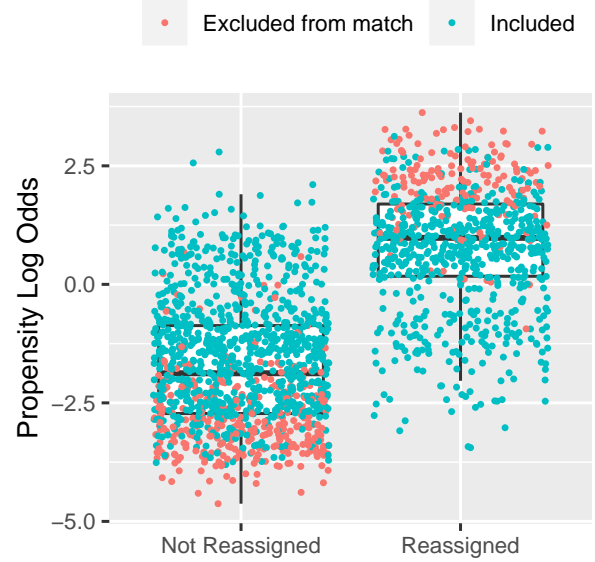


Figure 2: Estimated propensity scores for reassigned and not-reassigned students. Scores for students who were excluded from the ultimate match are colored red.

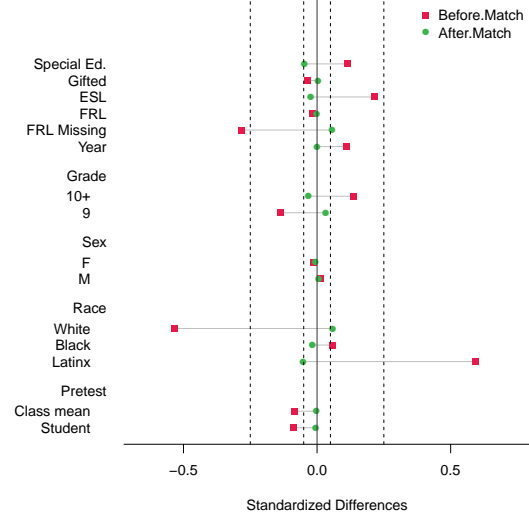


Figure 3: Covariate balance (standardized differences) before and after matching, for student level data. Dotted lines indicate standardized differences of 0.25 and 0.05, following the What Works Clearinghouse standards.

Table 2: Balance (standardized differences) on student level covariates before and after propensity score match. Omnibus p-values testing covariate balance are $p < 0.001$ before matching and $p = 0.95$ after matching.

	Before Match		After Match
	std.diff		std.diff
Pretest			
Class Mean	-0.09	.	0.00
Class Centered	-0.09	.	-0.01
Race/Ethnicity			
White	-0.53	***	0.06
Black	0.06		-0.02
Latinx	0.59	***	-0.05
Sex			
F	-0.01		-0.01
M	0.01		0.01
Grade			
10+	0.14	**	-0.03
9	-0.14	**	0.03
Special Ed.	0.12	*	-0.05
Gifted	-0.04		0.00
ESL	0.22	***	-0.02
FRL	-0.02		0.00
FRL Missing	-0.28	***	0.06
Year	0.11	*	0.00

`tools` package in R [3]. Before matching, several covariates were unbalanced, especially race. Table 2 shows stars reflecting p-values from individual covariate balance tests; nearly all covariates were unbalanced at the $\alpha = 0.1$ level. An omnibus balance test [12] gives $p < 0.001$. Figure 3 shows, as benchmarks, standardized differences of ± 0.25 and 0.5 , corresponding to thresholds given in the What Works Clearinghouse (WWC) handbook³ [5]. Before matching, imbalances in race and FRL missingness exceeded 0.25 , and most other imbalances were greater than 0.05 .

Matching improved nearly all of these imbalances. Most importantly, pretest measures were nearly exactly balanced. None of the individual covariate balance tests was significant at the 10% level or had standardized differences greater than 0.25 , and, with the exception of race, and FRL missingness none of the covariates was imbalanced with a standardized difference greater than 0.05 . The omnibus p-value testing overall balance was 0.95 .

The match also balanced classroom indicators. Before matching, the omnibus p-value testing balance of classroom indicators was < 0.001 ; after matching it was 0.99 .

6. THE EFFECT OF REASSIGNMENT

Table 3 gives five estimates for the effect of reassignment in classrooms where some students, but not all, were reassigned at some point. The first column gives the estimate itself, the second gives the sample size N for that estimate, the third, “Std Error” gives the standard error, and the fourth, “CI,” gives a 95% confidence interval. The last two columns contain sensitivity analyses, described in the following section. All the estimates used a regression routine from the `estimatr` package in R [2], with “HC2” heteroskedasticity-robust standard errors.

The first row, labeled “Raw,” is an unadjusted estimate, comparing all students in the sample who were reassigned to all students who weren’t. There is little difference in their average posttest scores.

The next row, labeled “Matched+Regression,” gives the effect estimate based on the match from Section 5. The lower sample size 1480 reflects the fact that some students were excluded from the match; this estimate only pertains to those who were included. To estimate the effect, we regress posttests on Z including a fixed effects for each match. Let $\hat{\tau}_m$ be the estimated effect in match m . If m is a pair—one reassigned student matched with one non-reassigned student—then $\hat{\tau}_m$ is the difference between the two students’ posttest scores. If there are more than two students in the match, $\hat{\tau}_m$ is difference in posttest means between reassigned and not-reassigned students within matched-set m . If treatment assignment is unconfounded within each match, $Z \perp \{y_C, y_T\} | \text{match}$, then $\hat{\tau}_m$ is unbiased for the average effect of Z on posttest scores in match m . Then the regression estimate is a weighted average of $\hat{\tau}_m$, with weights $w_m \propto (1/n_{1m} + 1/n_{0m})^{-1}$; this weighing scheme minimizes the standard error under standard linear regression assumptions (if the regressions assumptions do not hold, but Z is still unconfounded within the match, then the estimate is still unbiased but the weights are sub-optimal).

The next row, labeled “Match+Regression” uses the same regression model as the “Matched” estimator, but additionally controls for pretest scores (with a natural spline with five degrees of freedom), and indicators for special education status, missing free or reduced-price lunch data, and race. This strategy controls for differences in these covariates left over after the match, accounting for the fact that the match was imperfect.

The “Matched” and “Match+Regression” estimates were almost identical—effect sizes of -0.2 and -0.19 , respectively, with 95% confidence intervals of $[-0.29, -0.12]$ and $[-0.28, -0.11]$. These negative effect estimates suggest that reassignment hurts student learning. The effect size of a fifth of a standard deviation is roughly the same as the overall average effect of CTA1 in high schools in the second year of implementation, as estimated in [19], suggesting that reassignment may negate most of the positive effect of using CTA1.

The next two rows of Table 3, however, suggest that the

³In the context of a randomized experiment with attrition, covariate imbalances with standardized differences greater than 0.25 invalidate a study, whereas differences between 0.05 and 0.25 require statistical adjustment and differences less than 0.05 are acceptable as is.

	Estimate	N	Std. Error	CI	[Pretest]	[State]
Raw	-0.04	1981	0.03	[-0.11,0.03]	[-0.15,0.07]	[-0.16,0.08]
Matched	-0.20	1480	0.04	[-0.29,-0.12]	[-0.35,-0.06]	[-0.36,-0.05]
Match+Regression	-0.19	1480	0.04	[-0.28,-0.11]	[-0.33,-0.05]	[-0.34,-0.04]
Year 1	-0.24	1008	0.06	[-0.34,-0.13]	[-0.42,-0.06]	[-0.43,-0.04]
Year 2	-0.11	472	0.07	[-0.25,0.02]	[-0.33,0.11]	[-0.35,0.12]
Within-Class	-0.17	1981	0.04	[-0.24,-0.10]	[-0.29,-0.05]	[-0.30,-0.04]

Table 3: Estimates of the effect of reassignment without controlling for confounding (“Raw”), controlling for confounding with propensity score matching (“Matched”), with matching and further regression adjustment (“Match+Regression”), overall and separately for each year, and matching by classroom, with further regression adjustment (“Within-Class”). The table gives estimates, standard errors, 95% confidence intervals, and 95% sensitivity intervals assuming an unobserved confounder with properties similar to pretest scores (“[Pretest]”) and to State (“[State]”)

effect of reassignment may depend on context. Each row uses the “Match+Regression” approach, but separately in data from implementation years 1 and 2. It appears that reassignment may have hurt students’ posttest scores more in the first than in the second year of implementation—in the first year, we estimate an effect of -0.24 and in the second year we estimate an effect of -0.11. That said, the difference between the two effects is not itself statistically significant—that is, it may be the result of statistical noise.

The final row of Table 3, labeled “Within-Class,” uses a different confounder control strategy altogether. This estimate matches students by classroom, as if reassignment were randomized within classrooms. To weaken that assumption, the “Within-Class” estimate incorporates additional regression controls: a natural spline with five degrees of freedom for pretest, and indicator variables for the remaining covariates. This strategy estimates a similar negative effect as the others, NA, with a 95% confidence interval of [-0.24,-0.10].

6.1 Unobserved Confounding

The estimates in Table 3 all assumed (2), that there was no unobserved confounding. This assumption is strong, untestable, and could undermine all of the inference in Section 6. For instance, the estimated negative effect may be due to baseline differences in ability, beyond what is captured in pretest scores.

[15] suggest a method of estimating the sensitivity of a regression to an omitted confounder based on benchmarking from observed confounders. Roughly speaking, the idea is to widen the confidence interval from an ostensibly causal linear model to account for the possibility of a hypothetical unmeasured confounder, U , that predicts reassignment and posttests to the same extent as one of the observed covariates. These “sensitivity intervals” account for uncertainty from two sources: random error, and systematic error due to the omission of a confounder.

In order to confound the causal relationship between reassignment and posttests, a confounder would have to predict both. Capturing these two requirements, the method of [15] is based on two sensitivity parameters: first, T_Z encodes the extent to which U predicts Z , after accounting for observed covariates \mathbf{x} . Formally, T_Z is the t -statistic on the U coefficient from an ordinary least squares regression of Z on U and \mathbf{X} . The second parameter is ρ^2 , the squared partial

correlation between posttest scores and U , conditional on \mathbf{x} . Of course, since U is unobserved, neither T_Z nor ρ^2 is known; [15] suggest benchmarking them using observed covariates. That is, imagine each observed covariate, in turn, were unobserved, and calculate its T_Z and ρ^2 given the rest of the observed covariates.

Table 3 includes two such sensitivity intervals. The column labeled “[Pretest]” includes sensitivity intervals for an unobserved confounder that predicts reassignment and posttests as well as do pretest scores—typically the most important confounder. That is, these intervals are 95% confidence intervals that assume the possible existence of an unmeasured covariate as important as pretest. It turns out, in the current analysis, that omitting state indicators would cause more bias than omitting pretest scores; for that reason, the column labeled “[State]” gives sensitivity intervals for an unobserved confounder that predicts reassignment and posttest scores as well as state indicators. Both sets of sensitivity intervals are considerably wider than the corresponding confidence intervals, including both large and small negative effects. Sensitivity intervals for the “Matched”, “Match+Regression”, “Year 1,” and “Within-Class” estimates, whose confidence intervals excluded zero, excluded zero as well. That is, confounding from an unobserved variable as important as pretest or state may have led us to over-estimate the negative effect of reassignment; it may have also led us to under-estimate the effect. However, such confounding cannot explain the sign of the effect we estimated—even assuming the existence of an unobserved confounder as important as our most important covariates, the effect must be negative.

That said, an even stronger confounder, or more complex confounding from several unobserved covariates, may explain the observed results. Without a randomized trial, it is impossible to entirely rule out unobserved confounding.

6.2 Treatment Effect Heterogeneity

Previous research [16] has found evidence for a wide variety of uses for reassignment. In some cases, teachers reassign students who are falling behind their classmates, in other cases teachers reassign nearly the entire class to work on a particular section of the tutor, and in other cases teachers will simultaneously reassign all students working on a particular section *out* of that section. Along similar lines, our (inconclusive) evidence for variance between classrooms in

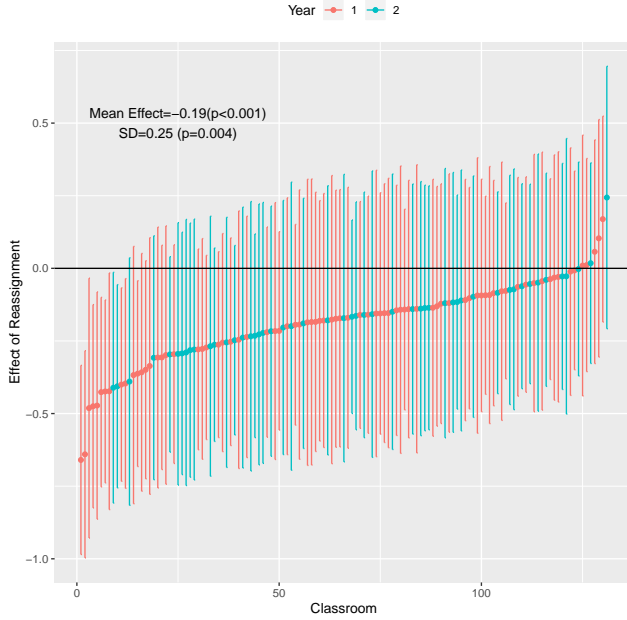


Figure 4: Classroom-specific effects of reassignment ($\hat{\beta}_1 + \hat{\gamma}_j$) from model (5). Error bars represent standard errors.

the relationship between pretest scores and the probability of a student being reassigned points towards varying uses for reassignment.

If reassignment is used differently from classroom to classroom, it stands to reason that it might have different effects in different classrooms, as well. To test that assumption, we fit a multilevel model with random effects for reassignment, varying by classroom. The model had the same fixed effects as model underlying the “Match+Regression” results described above, as well as random intercepts for classroom and random slopes for reassignment, varying by classroom. Formally, the model is:

$$\begin{aligned} Posttest_i = & \beta_{0,m[i]} + \beta_1 Z_i + \beta_2 SpEd_i + \\ & \beta_3 frlMIS_i + \beta_4 Black_i + \beta_5 Hisp_i + \\ & ns^5(pretest_i, \alpha) + \gamma_{j[i]} Z_i + \epsilon_{j[i]}^{Cls} + \epsilon_i^{Ind} \end{aligned} \quad (5)$$

where $\beta_{0,m[i]}$ is a fixed intercept for each match, $ns^5(pretest_i, \alpha)$ is a natural spline for pretest, with five degrees of freedom and coefficient vector α , and $\gamma_{j[i]}$, $\epsilon_{j[i]}^{Cls}$, and ϵ_i^{Ind} are random effects, modeled as normal with mean zero and standard deviation estimated from the data. Symbols α , β , γ , and ϵ do not represent the same quantities as in equation (4). $\gamma_{j[i]}$ is the random slope for reassignment, varying by classroom; the effect of reassignment in classroom j is estimated as $\hat{\beta}_1 + \hat{\gamma}_j$. That is, $\hat{\beta}_1$ represents the effect of reassignment, averaged over all classrooms, and γ_j represents the difference between classroom j ’s effect and the average. While precisely estimating the effect of reassignment in any particular classroom is beyond the scope of our data, this model allows us to estimate the variance across those effects, as the variance of γ_j s.

The results are displayed in Figure 4. The effect of reassignment in an average classroom is estimated as similar to the effects in Table 3. This effect varies with a standard deviation of approximately 0.25. To test for between-classroom variance, we compared the fit of the multilevel model to an analogous model without random slopes, with a likelihood ratio χ^2 test; the p-value was 0.004. This standard deviation is large enough to imply that the effect will be positive in some classrooms—indeed, Figure 4 shows a number of classrooms with positive effects. That said, the confidence intervals (based on estimates for the conditional variance of random slopes, combined with the standard error of the main effect of reassignment) are all rather wide and nearly all contain zero.

Therefore, while the effect of reassignment was negative, on average, it may have been positive in some classrooms.

This variation could be due to a number of factors, including differences in the composition of classrooms and in when or how reassignment is used. We considered two simple hypotheses about classroom-level predictors of heterogeneous treatment effects. The first hypothesis was that variance in students’ pretest scores within a classroom predicts the effect of reassignment in that classroom. The idea is that some teachers may use reassignment as a tool to address varying student ability—for instance, they may reassign lagging students to help them keep up with their classmates. Classrooms with higher variance in pretest scores afford more opportunities for teachers to use this reassignment strategy. If the strategy is widely used, and either particularly effective or ineffective at boosting students’ posttest scores, there will be a correlation between classroom-level variance in pretest scores and the effect of reassignment.

Our second hypothesis was that the proportion of students in a classroom who have been reassigned may predict classroom-level effects. The idea here is that in classrooms with a low proportion of students reassigned, teachers use reassignment in a more targeted fashion, so it may be more beneficial.

Figure 5 plots random effects $\hat{\gamma}_j$ from model (5) as a function of classroom level pretest variance and the proportion of students reassigned, respectively, with simple OLS fits. A positive relationship between pretest variance and $\hat{\gamma}_j$, and a negative relationship between proportion reassigned and $\hat{\gamma}_j$ are apparent, but with wide standard errors. To test these hypotheses more formally, we re-fit model (5), adding fixed effects for the variance in pretest scores and proportion reassigned, as main effects and interacted with Z_i . The model reduced the unexplained variance in classroom-level effects from 0.25 to 0.21—these variables explained about 27% of the unexplained variance in treatment effects. The coefficient on the interaction between pretest variance and reassignment—measuring the extent to which pretest variance explains treatment effects—was estimated as 0.09, with a 95% confidence interval of $[-0.75, 0.93]$, so the data are compatible with large associations in either direction between pretest variance and treatment effects. No firm conclusions may be drawn. The coefficient on the interaction between proportion reassigned and reassignment—measuring the extent to which classroom proportion reassigned ex-

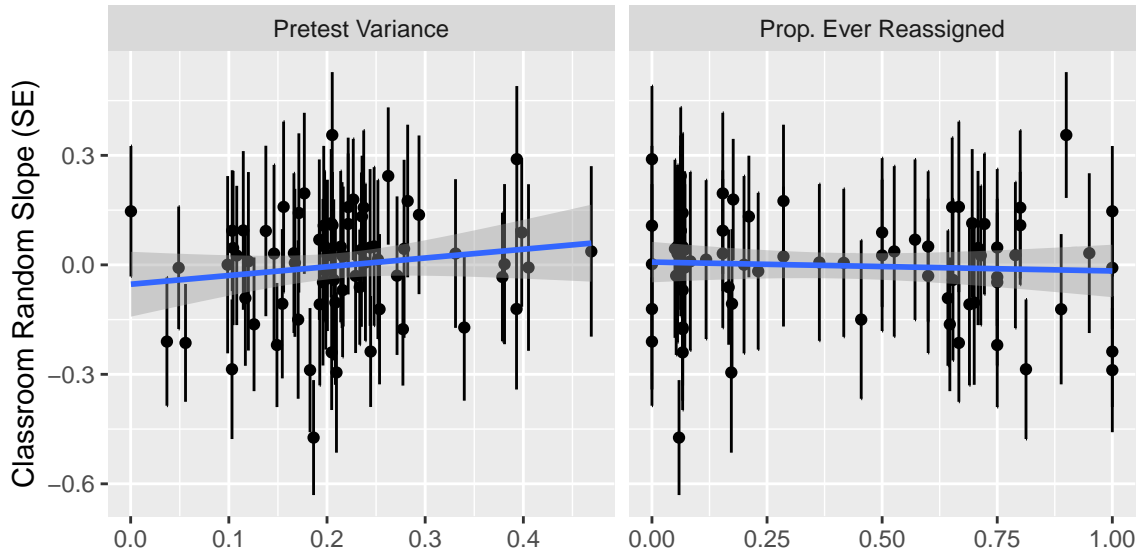


Figure 5: The random effects γ_j from model (5) (with error bars for one standard error) as a function of classroom-level variance in pretest scores and the proportion of students in a classroom who were ever reassigned. OLS fits are added for interpretation.

plains treatment effects—was estimated as -0.3, with a 95% confidence interval of [-0.57,-0.03], and a p-value of $p = 0.03$. This suggests that the effect of reassignment may be lower—more negative—in classrooms in which a higher proportion of students were reassigned. This aligns with our second hypothesis.

These effect heterogeneity analyses assume (2), no unmeasured confounding. Unfortunately, we are not aware of methods for sensitivity analysis of the type presented in Section 6.1, applied towards estimates of effect heterogeneity. In particular, unobserved confounding may vary by classroom; for instance, the structure of the propensity score match may vary with the proportion of students ever reassigned, since within-classroom matches will be scarce when this proportion is high. For those reasons, the conclusions in this section should be taken as suggestive and exploratory.

7. DISCUSSION

A deeper understanding of the use of reassignment and its effects can yield practical and theoretical dividends. Teachers would benefit from clear guidelines as to when and whether reassigning students to a new section may benefit that student’s learning. A better understanding of if and when reassignment helps or hurts student learning can contribute to our understanding of the importance of sequence and mastery learning in intelligent tutoring systems.

Here, we estimate that, on average, reassignment hurts student learning, perhaps as much as CTA1 helps. That conclusion comes with two important caveats: first, although it appears unlikely that the entire reassignment effect we estimated is due to confounding from unmeasured variables, a large portion of the effect might be. That is, the magnitude of the reassignment effect we estimated may be an artifact of unmeasured confounding—reassignment may not be as bad

as we estimate, or it may be worse. (Of course, we cannot rule out that the entire effect is due to confounding, or that the direction of our estimated effect is wrong.)

Secondly, there is evidence that the effect of reassignment varies widely between classes. Even if it hurts on average, used properly it may help.

More broadly, these issues illustrate the opportunities and perils of analyses of log data from randomized trials of educational technology. Even when the randomization itself does not contribute to an analysis, the combination of log data collected under natural conditions and a long period of time and a posttest measuring student ability at the end of the study can be used to gain insights on tutor use and effects. On the other hand, log data, even from a randomized trial, is observational, and therefore messy and subject to confounding and other threats. Causal modeling of log data from randomized experiments is crucial, but difficult.

8. ACKNOWLEDGMENTS

This work was partially funded by NSF grant #DRL-1420374.

9. REFERENCES

- [1] D. Bates, M. Mächler, B. Bolker, and S. Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015.
- [2] G. Blair, J. Cooper, A. Coppock, M. Humphreys, and L. Sonnet. *estimatr: Fast Estimators for Design-Based Inference*, 2019. R package version 0.20.0.
- [3] J. Bowers, M. Fredrickson, and B. Hansen. *RIttools: Randomization Inference Tools (Development Version)*, 2017. R package version 0.2-0.
- [4] M. Caliendo and S. Kopeinig. Some practical guidance for the implementation of propensity score matching.

- Journal of economic surveys*, 22(1):31–72, 2008.
- [5] W. W. Clearinghouse. Standards handbook, version 4.1, 2020.
 - [6] T. D. Cook, W. R. Shadish, and V. C. Wong. Three conditions under which experiments and observational studies produce comparable causal estimates: New findings from within-study comparisons. *Journal of Policy Analysis and Management: The Journal of the Association for Public Policy Analysis and Management*, 27(4):724–750, 2008.
 - [7] T. D. Cook, P. M. Steiner, and S. Pohl. How bias reduction is affected by covariate choice, unreliability, and mode of data analysis: Results from two types of within-study comparisons. *Multivariate Behavioral Research*, 44(6):828–847, 2009.
 - [8] A. T. Corbett, K. R. Koedinger, and W. Hadley. Cognitive tutors: From the research classroom to all classrooms. *Technology enhanced learning: Opportunities for change*, pages 235–263, 2001.
 - [9] S. Doroudi, K. Holstein, V. Aleven, and E. Brunskill. Sequence matters but how exactly? a method for evaluating activity sequences from data. *Grantee Submission*, 2016.
 - [10] A. Gelman and J. Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press, 2006.
 - [11] B. B. Hansen. Full matching in an observational study of coaching for the sat. *Journal of the American Statistical Association*, 99(467):609–618, 2004.
 - [12] B. B. Hansen and J. Bowers. Covariate balance in simple, stratified and clustered comparative studies. *Statistical Science*, 23(2):219–236, 2008.
 - [13] B. B. Hansen and S. O. Klopfer. Optimal full matching and related designs via network flows. *Journal of Computational and Graphical Statistics*, 15(3):609–627, 2006.
 - [14] T. J. Hastie. Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge, 2017.
 - [15] C. A. Hosman, B. B. Hansen, P. W. Holland, et al. The sensitivity of linear regression coefficients’ confidence limits to the omission of a confounder. *The Annals of Applied Statistics*, 4(2):849–870, 2010.
 - [16] A. Israni, A. C. Sales, and J. F. Pane. Mastery learning in practice: A (mostly) descriptive analysis of log data from the cognitive tutor algebra i effectiveness trial, 2018.
 - [17] K. R. Koedinger. *Toward evidence for instructional design principles: Examples from Cognitive Tutor Math 6*. ERIC Clearinghouse, 2002.
 - [18] J. Neyman. Sur les applications de la théorie des probabilités aux expériences agricoles: Essai des principes. *Roczniki Nauk Rolniczych*, 10:1–51, 1923.
 - [19] J. F. Pane, B. A. Griffin, D. F. McCaffrey, and R. Karam. Effectiveness of cognitive tutor algebra i at scale. *Educational Evaluation and Policy Analysis*, 36(2):127–144, 2014.
 - [20] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.
 - [21] F. E. Ritter, J. Nerb, E. Lehtinen, and T. M. O’Shea. *In order to learn: How the sequence of topics influences learning*, volume 2. Oxford University Press, 2007.
 - [22] S. Ritter, A. Joshi, S. Fancsali, and T. Nixon. Predicting standardized test scores from cognitive tutor interactions. In *Educational Data Mining 2013*, 2013.
 - [23] P. Rosenbaum. *Observational studies*. Springer, 2002.
 - [24] P. Rosenbaum and D. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.
 - [25] D. B. Rubin. Bayesian inference for causal effects: The role of randomization. *The Annals of statistics*, pages 34–58, 1978.
 - [26] A. Sales, T. Patikorn, and N. Heffernan. Bayesian partial pooling to improve inference across a/b tests in edm. In *Proceedings of the 11th International Conference on Educational Data Mining. International Educational Data Mining Society*, 2018.
 - [27] A. C. Sales, B. B. Hansen, and B. Rowan. Rebar: Reinforcing a matching estimator with predictions from high-dimensional covariates. *Journal of Educational and Behavioral Statistics*, 43(1):3–31, 2018.
 - [28] A. C. Sales, J. F. Pane, et al. The role of mastery learning in an intelligent tutoring system: Principal stratification on a latent variable. *The Annals of Applied Statistics*, 13(1):420–443, 2019.
 - [29] R. Scheines, E. Silver, and I. M. Goldin. Discovering prerequisite relationships among knowledge components. In *EDM*, pages 355–356, 2014.
 - [30] W. R. Shadish and P. M. Steiner. A primer on propensity score analysis. *Newborn and Infant Nursing Reviews*, 10(1):19–26, 2010.
 - [31] A. Vuong, T. Nixon, and B. Towle. A method for finding prerequisites within a curriculum. In *EDM*, pages 211–216, 2011.
 - [32] H. Wickham, M. Averick, J. Bryan, W. Chang, L. D. McGowan, R. François, G. Grolemond, A. Hayes, L. Henry, J. Hester, M. Kuhn, T. L. Pedersen, E. Miller, S. M. Bache, K. Müller, J. Ooms, D. Robinson, D. P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo, and H. Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019.
 - [33] D. Wood, J. S. Bruner, and G. Ross. The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, 17(2):89–100, 1976.
 - [34] Y. Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2020. R package version 1.28.

Feature Selection Metrics: Similarities, Differences, and Characteristics of the Selected Models

Debopam Sanyal
University of Illinois at
Urbana-Champaign
Champaign, IL, USA
dsanyal2@illinois.edu

Nigel Bosch
University of Illinois at
Urbana-Champaign
Champaign, IL, USA
pnb@illinois.edu

Luc Paquette
University of Illinois at
Urbana-Champaign
Champaign, IL, USA
lpaq@illinois.edu

ABSTRACT

Supervised machine learning has become one of the most important methods for developing educational and intelligent tutoring software; it is the backbone of many educational data mining methods for estimating knowledge, emotion, and other aspects of learning. Hence, in order to ensure optimal utilization of computing resources and effective analysis of models, it is essential that researchers know which evaluation metrics are best suited to educational data. In this article, we focus on the problem of wrapper feature selection, where predictors are added to models based on how much they improve model accuracy in terms of a given metric. We compared commonly-used machine learning algorithms including naive Bayes, support vector machines, logistic regression, and random forests on 11 diverse learning-related datasets. We optimized feature selection based on nine different metrics, then evaluated each to address research questions about how effective each metric was in terms of the others (e.g., does optimizing for precision also result in good F1?) as well as calibration (i.e., are predictions produced by models accurate probabilities of correctness?). We provide empirical evidence that the Matthews correlation coefficient (MCC) produced the overall best results across the other metrics, but that root mean squared error (RMSE) selected the best-calibrated models. Finally, we also discuss issues related to the number of features selected when optimizing for each metric, as well as the types of datasets for which certain metrics were more effective.

Keywords

Feature selection, Metrics, Machine learning, Student models

1. INTRODUCTION

Machine learning is a popular method for building predictive models that automatically estimate various aspects of learning. These models, in turn, can be applied to study the processes of learning or teaching, or to automatically

guide students as they learn. Training models is a complex process, however. The space of possible machine learning models is far too large to fully explore, and thus the search space is typically narrowed by focusing on candidate models that appear promising via some measure of correctness (agreement with ground truth labels, for supervised classification), such as Cohen's kappa or F_1 [16, 40]. One common methodological step that involves model selection (narrowing the search space) is *wrapper forward feature selection* [29], a process wherein features are added one at a time to a model based on which feature produces the largest gain in model correctness. Changing the correctness metric by which features are evaluated can have a significant impact on the final selected model (which we demonstrate in this paper); however, little is known about exactly what these impacts are for different correctness metrics. In this paper, we address this problem by performing feature selection based on different metrics and comparing the resulting models.

Previous work in the area of examining correctness metrics for educational data mining has largely focused on what those metrics reveal about models [40, 10]. Related work has shown, for example, that area under the receiver operating characteristic curve (AUC or AUROC) ignores the scale of model predictions [40], and that F_1 can be increased by over-predicting the positive class [10]. From such findings we can generate hypotheses about the properties of models that result from relying on those metrics during feature selection. For example, we might expect recall- and F_1 -based feature selection to favor models that over-predict the positive class. However, there is little empirical evidence to support such hypotheses, which we aim to provide in this paper.

We explore a wide variety of correctness metrics for feature selection, evaluating them on 11 education-related datasets, to empirically measure relationships between feature selection metrics and resulting models. We include well-known and extensively-used metrics like AUC, Cohen's kappa, and others, as well as metrics that are less-commonly used but perhaps equally valuable, like the Matthews correlation coefficient and the minimum proper AUC. We experiment with metrics and datasets across four commonly-used machine learning classifiers, including support vector machine, naive Bayes, logistic regression and random forest. These algorithms have been frequently applied with great success in educational data mining and related research [24, 21, 43, 9], including in situations where high-dimensional data require feature selection [27, 49, 34].

Debopam Sanyal, Nigel Bosch and Luc Paquette "Feature Selection Metrics: Similarities, Differences, and Characteristics of the Selected Models" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 212 - 223

To the best of our knowledge, ours is the first work to explicitly test differences between correctness metrics in the context of feature selection. Our results are valuable for future educational data mining research and practice by providing guidance to machine learning experts who wish to make evidence-based decisions about their model building methods. In particular, we characterize metrics in terms of the models that result from performing feature selection based on those metrics, which will help researchers decide on appropriate metrics based on the desired properties of their resulting models.

2. RELATED WORK

While previous research and other projects in this area is limited, there have been a few relevant research projects with findings that significantly informed our current work. In this section, we describe metrics evaluated in this study along with examples where they were used in previous work, then discuss directly-related work on evaluating metrics in educational data mining.

2.1 Metrics and their Usage

Accuracy. In this paper, accuracy refers to the proportion of correctly classified instances, though in other contexts it may refer more generally to any measure of how well a model's predictions align with ground truth values. Accuracy is one of the most straightforward metrics to calculate and understand, and thus has been reported frequently in machine learning studies [35, 12]. However, previous research has noted flaws with accuracy. In situations where labels are imbalanced, accuracy is often attenuated [25] or inflated [10] depending on the rate at which the model predicts the majority class. Despite possible flaws, it is commonly examined and is often the default correctness measure in machine learning software [39], including in wrapper feature selection software [41], so we include it in this paper.

AUC. AUC measures model correctness in terms of true positive rate across every possible false positive rate (i.e., across all possible decision thresholds). Chance level AUC is 0.5, while a perfect model has $AUC = 1$ and a completely incorrect model has $AUC = 0$. AUC is a valuable metric for its clear interpretability and effectiveness in the face of class imbalance [25], and has often been reported as an evaluation metric on educational datasets (e.g., [26, 23, 40, 37]). However, it only measures correctness in terms of the order of predicted values, not their scale [40], so it is unclear whether selecting features based on AUC will result in models that may have poorly-scaled predictions (an issue we explore in this paper). A related metric is the area under the precision–recall curve (AUPRC) [44], which also considers all possible decision thresholds. We have not yet included AUPRC in analyses, but expect that its behavior with respect to scale of predictions may be similar to AUC.

MPAUC. In situations where models provide only binary predictions, an approximation of AUC can be calculated by measuring the minimum proper AUC (MPAUC) of the quadrilateral formed by the single available decision threshold [38], as shown in Figure 1. We refer to this metric as MPAUC for the sake of brevity when reporting results, though it is not typically abbreviated in previous literature. It differs from AUC in that it measures the area for a “curve”

defined by a single point instead of many points as in AUC. Its advantage is that it is applicable even when continuous decision thresholds are not available. MPAUC has been utilized as a metric for feature selection in prior educational data mining research [9], but it is unclear how it compares to alternatives we explore in this paper.

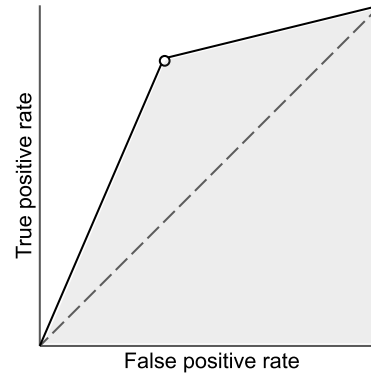


Figure 1: Example MPAUC (shaded area).

MCC. The Matthews Correlation Coefficient (MCC) measures the correlation between two binary variables (predicted labels and actual labels) [30], and is equivalent to Pearson's r for two binary variables (i.e., ϕ). MCC ranges from -1 to 1, where 0 indicates chance level and 1 indicates perfect classification. MCC is especially useful in binary classification models where there is class imbalance, since its chance level is not affected by imbalance. MCC is simply a correlation coefficient between the true and predicted class. It is only defined for binary variables. While it is not common in educational data mining research, it has been occasionally reported [8, 1] and is valued in other machine learning fields [15].

Recall. Recall is the proportion of a certain label class (typically the positive class) that was correctly identified as being in that class [46, 4]. Recall is an informative measure for understanding model correctness, especially in situations where it is important to focus on one class (e.g., in situations where false negatives are costly). However, it can be inflated by over-predicting the positive class [10] and is thus not often reported as the sole measure of model correctness, so it is unclear whether it is appropriate as a metric for feature selection.

Precision. Precision is similar to recall; it is the proportion of instances predicted as being in the positive class that were correct predictions. Like recall, it is typically only reported in conjunction with other correctness metrics, but unlike recall it cannot be inflated by over-predicting the positive class [10]. However, in some cases it can be maximized by predicting the positive class for only a few of the highest-confidence instances.

F₁. F₁ is defined as the harmonic mean of precision and recall, and thus avoids some of issues of recall (favoring over-prediction of the positive class) and precision (under-predicting the positive class). However, it can be inflated by over-predicting the positive class [10], so it is unclear

whether selecting features based on F_1 will favor models that over-predict the positive class or not.

RMSE. RMSE (root-mean-square error) measures the Euclidean distance between predictions and ground truth labels. Since RMSE is an error metric, lower values are better, with 0 indicating no error. It is commonly associated with regression problems, since it can be easily calculated for continuous labels, but is also effective for binary classification with models that produce continuous-valued probability predictions [40, 13]. Previous research has noted that RMSE is especially effective for optimizing probabilistic predictions [40]; thus, we expect that selecting features based on RMSE might also produce models with well-calibrated probabilities (where model confidence matches the probability that the model is correct). Like AUC, RMSE does not require setting a decision threshold, unlike the other metrics we consider in this study. We refrained from using close variants like Mean Absolute Deviation (MAD) or Error (MAE), since previous work has noted issues with these metrics for model selection [40].

Kappa. Cohen’s kappa (κ) was developed as a measure of agreement between human annotators [16], but has often been utilized as a machine learning correctness metric by measuring the agreement between ground truth labels and predicted labels [10]. Like correlation measures, kappa ranges from -1 to 1 where 0 is random chance and 1 indicates perfect classification.

2.2 Research on Metrics in Educational Data Mining

Previous research has focused on metrics primarily in terms of the perspective that metrics have on a model of students, or on the properties of the model that are highlighted (or hidden) by particular metrics.

In one previous project, researchers focused on evaluating the properties of metrics that require continuous (probability-like) predictions [40]. In particular, they focused on AUC, RMSE, mean absolute error (MAE), and log likelihood (LL). They noted that for some applications (e.g., prediction of probability that a student has mastered a specific skill) metrics such as AUC do not favor well-calibrated models. They also compared metrics in terms of how often they agreed on picking the best model out of a pool of 20 simulated datasets, finding that RMSE and LL frequently agreed (17 out of 20) but others agreed much less often; the second-highest agreement was between RMSE and AUC, on 7 out of 20 datasets. This is especially relevant to the work in this paper, where we compare properties of metrics applied across 11 real-world datasets.

In similar previous work, researchers compared the properties of metrics that require binary or categorical predictions, rather than continuous predictions [10]. They noted that F_1 is influenced by the base rate of the positive class in data, in line with other research on Cohen’s kappa, AUC, and other metrics [25]. However, they also noted that F_1 (and recall) are influenced by the predicted rate of classifiers. This finding is especially relevant to the current research because it is possible that feature selection will favor models and features that tend to predict more of the positive class when

selecting based on these metrics.

3. FRAMING THE PROBLEM

The goal of this paper is, broadly speaking, to provide empirical results that illustrate the relationships 1) among different metrics, and 2) between metrics and models, when metrics are employed for forward feature selection.

Sequential feature selection is a type of wrapper (model-based) feature selection in which a feature is added to or removed from a model, the model is re-trained, and the quality of the feature in question is assessed based on improvement in model correctness (as measured by some metric). In this study, we specifically performed forward feature selection by adding one feature at a time, stopping when all features were added or when the model had not improved for three consecutive features, then returned the set of features with maximum correctness among all the combinations explored. Our work focuses primarily on the effects of utilizing different metrics for the step in which model correctness is assessed, which drives the entire feature selection process. We define four research questions (RQs) to explore this problem:

RQ1: When selecting features based on a specific metric, how do the results vary in terms of the other metrics? Addressing this question will inform decisions about which metric to apply during feature selection by showing the relationships between metrics. For example, some low-cost applications may benefit from high recall (e.g., automatically selecting the most relevant material for students to review) while other higher-cost applications may require high precision (e.g., automatically predicting when a teacher should intervene to redirect learning behaviors). In these examples, we may wish to optimize feature selection for different metrics, but it is crucial to understand how that might influence other metrics; e.g., does optimizing feature selection for AUC tend to produce models that are also good in terms of Cohen’s kappa, recall, and the other metrics?

To address RQ1 we define the *ranking* of a metric with respect to all the other metrics. Specifically, given a set of metrics \mathcal{M} , a selection metric $X \in \mathcal{M}$ has rank 0 with respect to another metric $Y \in \mathcal{M}$ if selecting features based on X results in the best¹ value of Y compared to selecting features based on all other metrics in \mathcal{M} . Likewise, a metric $Z \in \mathcal{M}$ has rank 1 with respect to Y if selecting features based on Z produces the second-best value of Y compared to all other metrics in \mathcal{M} , and so on. Generally, we expect that selecting features for some metric $X \in \mathcal{M}$ will have rank 0 with respect to itself (X), though this is not necessarily always true. Furthermore, some metrics may be generally better than others in terms of rank, if they tend to favor models with well-rounded properties that satisfy each metric. We thus calculate the *mean ranking* of each metric as the mean of all rankings for a metric with respect to itself and all other metrics (nine in total, in this paper), as a way to discover which feature selection metrics tend to yield models that satisfy the wide range of criteria imposed by different metrics.

¹“Best” meaning highest for most metrics, but lowest for RMSE since it is an error metric.

RQ2: How do different feature selection metrics impact model calibration? As previous work noted, some metrics do not penalize models for being poorly calibrated [40]. However, it remains unclear how large of an effect using different metrics during feature selection may have on the calibration of the resulting model. We address this research question by calculating CAL scores (described in Sec. 4.4) for models selected based on each metric [12].

RQ3: How do different feature selection metrics impact the predicted rates of models? Certain correctness metrics favor over- or under-prediction of the positive class more than others. For example, accuracy for a problem with imbalanced classes can be increased simply by biasing predictions of the positive class in the same direction as the imbalance in the data [10]. We might expect that relying on accuracy for feature selection could thus result in models that over or under-predict the positive class, but it is unclear how problematic these effects may be, which we measure in addressing this research question.

RQ4: Do some feature selection metrics tend to result in more parsimonious models (fewer features) than others? In addressing this research question, we further characterize the models that result from applying different metrics during feature selection, and highlight cases where feature selection may fail (by selecting too few features) or unnecessarily increase model complexity (by selecting an unusually large number of features).

4. EXPERIMENTS

We performed a variety of experiments to address our research questions, consisting of training and testing machine learning classifiers with forward feature selection. Experiments required approximately 11 months of continuous run time², given that we performed extensive hyperparameter selection with 4 classifiers, 11 datasets, and 9 feature selection metrics, as detailed in this section.

4.1 Classifiers

As mentioned in the Introduction, we trained models including random forest, support vector machines, naive Bayes, and logistic regression. These machine learning algorithms represent a variety of methods with differing assumptions and levels of flexibility, and which are frequently employed in educational data mining research [18, 5, 21, 43, 20, 7, 11, 45]. Moreover, with the possible exception of random forest, these models quite often benefit from feature selection to avoid problems of over-fitting (e.g., when a logistic regression has nearly as many parameters as instances) [33] and collinearity (e.g., when two very similar features incorrectly double the impact of a relationship in a naive Bayes model).

4.2 Cross-validation

We utilized student-level four-fold cross-validation, training each model on data from 75% of students and testing it on the remaining 25% of students, then repeating a total of four times until each student was in the testing data exactly once. This procedure ensured that data from the same student was

²Experiments were run on an Intel Core i7 4.2 GHz processor (using a single core) with 32 GB memory and 256 GB storage.

never present in training and testing at the same time, which was crucial given that some of our datasets had multiple instances per student.

We performed nested (within training data) student-level four-fold cross-validation for evaluating hyperparameters and selecting features. Specifically, for every possible combination of hyperparameters, we performed forward feature selection, then stored the best result from the feature selection process (according to the current selection metric). Finally, we retrained the model using the best set of hyperparameters, including the best features, on all training data, and applied it to the testing data. Hyperparameter selection and feature selection did not involve the testing set in any way.

There are two common strategies for evaluating the results of cross-validation. The first, *macro-level averaging*, consists of calculating the desired correctness metric for each fold and averaging across folds (four folds, in our case). The second strategy, *micro-level averaging*, involves storing the predictions of each fold and calculating the correctness metric once at the end based on all predictions. We evaluated both strategies to assess possible differences on the feature selection process.

4.3 Hyperparameters

We extensively tested common hyperparameters for each classification algorithm to ensure models had a chance to fit to the very different properties of our datasets (e.g., type of data, number of features, size of dataset).

For random forest we set the number of trees at 50 (significantly increasing this proved infeasible for an already-long run time). We varied the minimum number of samples required to create a branch in each tree, trying 5 different values (2, 4, 8, 16, or 32). This hyperparameter controls model complexity by restricting how fine-grained the decisions in each tree can be. We also varied the number of features randomly chosen for building each tree, testing 4 options including proportions of .25, .50, .75, and the square root of the number of features (the default setting). This hyperparameter controls how different trees are from each other in terms of the features from which they are trained. In total, there were $5 \times 4 = 20$ combinations of hyperparameters for random forest.

We trained SVMs with the radial basis function (RBF) kernel, which has a hyperparameter γ that controls the size (radius of influence) of each RBF kernel. We tried values for γ of 0.001, 0.01, 0.1, 1, and 10. Similarly, we tuned C , the SVM complexity hyperparameter, over the same set of 5 possible values. There were thus $5 \times 5 = 25$ hyperparameter combinations for SVM.

Naive Bayes has little in the way of hyperparameters to tune, apart from the distribution assumption to use. We assumed a Gaussian distribution for all models, and thus did not perform grid search across hyperparameters.

We trained logistic regression models with L_2 regularization, and tuned the strength of regularization as a hyperparameter over the space of 5 possible values: 0.001, 0.01, 0.1, 1, and 10.

Finally, we experimented briefly with hyperparameters related to class imbalance in the datasets, after noting that models frequently learned to only predict the majority class. We initially experimented with re-weighting instances of the minority class with higher weight set as a hyperparameter, but ultimately found that generating synthetic minority-class data via SMOTE (Synthetic Minority Over-sampling TEchnique [14]) was more consistently effective across our datasets without requiring hyperparameter tuning.

4.4 Measuring Model Calibration

Calibration refers to how well a model’s predicted probabilities match the probability that those predictions are correct. For example, given a set of 100 instances where model predictions are all ≈ 0.7 , we would expect 70 of the instances to be the positive class, and 30 to be in the negative class. If more than 70 are true positives, the model is under-confident for those 100 instances, while if fewer are true positives, the model is overconfident. Good model calibration is desirable so that predictions are interpretable as probabilities, allowing decision thresholds to be set in meaningful ways (e.g., triggering an intervention only if the model is at least 90% confident, knowing that it will thus result in a 10% false positive rate).

We measured calibration by calculating CAL scores [12]. The CAL score for a model is calculated by sorting all N instances according to predicted probability, then dividing into $N - 99$ sliding windows of 100 instances (sliding by 1 instance). For each window, we calculated the absolute difference between the base rate of the positive class for those 100 instances and the mean predicted probability for the same instances. The CAL score consists of the mean of those absolute differences across all windows, and can be interpreted as the mean absolute error in model confidence.

4.5 Datasets

4.5.1 Video-based Engagement Detection Datasets

We obtained six datasets from a study that measured students’ self-reported engagement during an essay writing task [31], during which students’ faces were recorded by a video camera. Students made verbal judgments of their engagement in the moment (concurrently) in response to auditory probes. One week later, they made retrospective judgments of their engagement by viewing video clips of themselves that were recorded during the essay writing task. There were 23 students who made a total of 530 judgments of engagement during the writing task and 1,325 retrospective judgments. Researchers extracted three sets of features from videos: 1) heart rate, estimated via photoplethysmography [32]; 2) animation units (ANUs), a set of facial feature descriptors provided by the Microsoft Kinect SDK, which are analogous to facial action units (AUs) [19]; and 3) local binary patterns in three orthogonal planes (LBP-TOP) [50], which capture facial textures and how those textures change over time.

There were thus two sets of labels and three sets of features, for a total of six video-related datasets. We refer to the two heart rate datasets as VIDEO-HR-C (concurrent labels) and VIDEO-HR-R (retrospective labels). Similarly, we refer to the two animation unit datasets as VIDEO-ANU-C and VIDEO-ANU-R, and the two LBP-TOP datasets as VIDEO-LBP-C and VIDEO-LBP-R.

4.5.2 Cognitive Tutor Algebra Datasets

We obtained two datasets from a study [36] in which 59 students interacted with a computerized learning environment called Cognitive Tutor Algebra [3]. Students used Cognitive Tutor Algebra for an entire year as part of their regular mathematics curriculum. Researchers labeled 10,397 sequences of student actions in the learning environment for the presence of “gaming the system” behavior, where students attempt to progress through material by exploiting features of the learning environment (e.g., requesting hints repeatedly, guessing many answers) [6].

Researchers extracted two sets of features. Pattern features captured the presence or absence of 60 different sequences of actions that were designed to be similar to patterns identified by domain experts. We refer to the dataset with pattern features as CTA-PF in this paper. The second set of features consisted of 25 count features. Count features captured the number of times 6 different actions occurred as well as the number of times 19 different events occurred. Events were identified by domain experts, and included things like pausing between attempts to answer a problem or trying to reuse an answer in multiple steps of a problem. We refer to the dataset with 25 count features as CTA-C in this paper.

4.5.3 Student Survey Datasets

Two additional datasets came from surveys obtained from 788 students at two different secondary schools during the 2005–2006 school year [17]. The survey consisted of 30 questions, including demographics, which school they attended (of two possibilities), and other variables. We one-hot encoded variables with categorical answers. Labels in both datasets consisted of course grades recorded on a 0–20 scale. We converted these to binary labels by splitting on the median into high and low grades, so that all datasets would be comparable binary classification problems.

One of the datasets came from students in a mathematics course (MATH, with 395 students) and the other from a Portuguese language class (PORTUGUESE, with 649 students). Some students were in both classes; thus, the total number of students was less than the sum of the classes.

4.5.4 Educational Process Mining Dataset

We also extracted features from an educational process mining (EPM) dataset. Students worked on electronics exercises in a software environment called DEEDS (Digital Electronics Education and Design Suite). Students’ actions in the learning environment were timestamped and logged, and included mouse movements, keystrokes, and information about the exercises being solved. Grade data were provided for five learning sessions, from which we extracted features including time spent on activities, number of actions, mean, standard deviation, and other summary features from problem-level data. In total, 115 students participated, but grades and action log data were not available for all students in every session. Grades were recorded on a numeric scale, though we again converted these to classification problems via median split to maintain consistency with other datasets.

5. RESULTS AND DISCUSSION

We focus results on the four research questions outlined in Section 3; we also provide model correctness results in the

Appendix, but do not focus on these results here since the goal of this work is to compare metrics rather than focus on improving over previously-published models. Our experiments to address the research questions included 4 different machine learning algorithms, 2 methods of calculating results during cross-validation, and 11 datasets. The different machine learning algorithms yielded similar patterns for our primary research question (RQ1), with only a few exceptions (Figure 2). Similarly, results differed little across macro- and micro-averaging methods (Figure 3). Thus, we aggregated across classification algorithms and averaging methods to address our research questions without unnecessarily dividing results into 8 (2 averaging levels \times 4 classifiers) subsets.

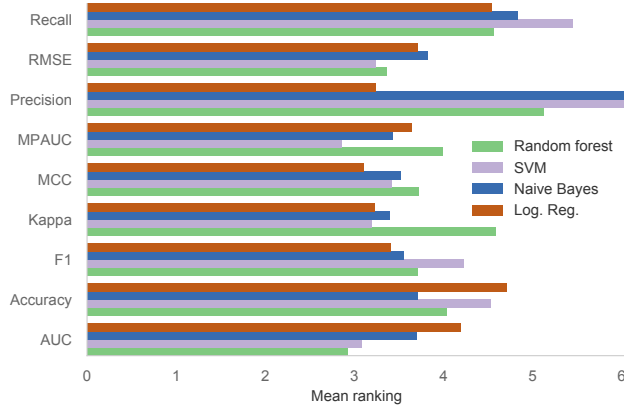


Figure 2: Mean ranking for each machine learning algorithm and feature selection metric. “Log. Reg.” refers to logistic regression.

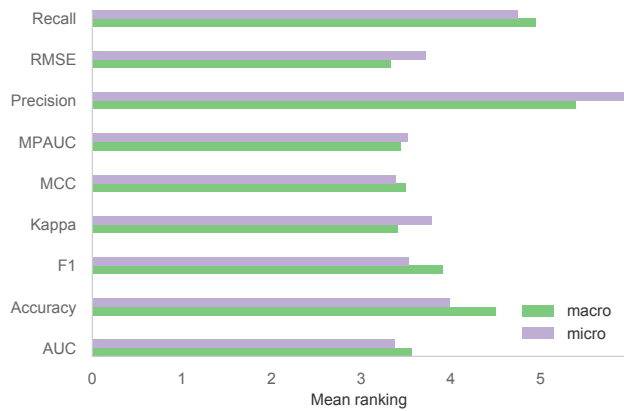


Figure 3: Mean ranking for feature selection metrics when calculating results via macro-level versus micro-level averaging.

5.1 Mean Rankings

RQ1 asks *When selecting features based on a specific metric, how do the results vary in terms of the other metrics?* Results in Table 1 show that MCC was, on average, the best (lowest) across models and datasets. Mean ranking for MCC averaged 3.441 across all datasets, while AUC and MPAUC were similar with mean rankings of 3.468 and 3.476 respectively. Low rank for MCC indicates that, across 11 datasets,

selecting features based on improvement in MCC yielded better results (in terms of itself and the other 8 metrics) than selecting features based on any of the other metrics. Specifically there were 3.441 correctness metrics on average for which selecting features based on some metric other than MCC yielded better results than MCC.

Conversely, precision was the worst-performing metric in terms of producing good results for other correctness metrics, with a mean ranking of 5.672. Recall and accuracy both had mean rankings above 4, while all other selection metrics had rankings \approx 3.5.

There was also some notable variation across datasets. Possible causes of variations include the differing types of features in the datasets (binary, continuous, counts, etc.), class imbalance, and problem difficulty (e.g., signal to noise ratio). A handful of datasets had significantly lower mean rank values for a specific metric when compared other metrics and the average value across all datasets for the metric itself. For example, in the PORTUGUESE dataset, AUC was a particularly effective metric. AUC’s mean ranking was 1.764, indicating that selecting features based on AUC in that dataset was almost always better (in terms of itself and the other metrics) than optimizing for those metrics was. In other datasets like VIDEO-LBP-C, the best metric had a much higher mean ranking. Similarly, metrics like F_1 and Accuracy had unusually low mean rank values for the MATH and VIDEO-HR-R datasets, respectively. In such cases, one metric did not frequently outperform the others.

We also explored RQ1 visually by counting the number of datasets for which each metric had at least a certain ranking or better (Figure 4), much like constructing a receiver operating characteristic curve requires finding predictions above every possible threshold. In Figure 4, higher curves are better, indicating that there were more datasets where the metric had a desirable ranking. The curve for precision was clearly lowest, followed by recall and then accuracy. The rest of the metrics were similar to one another, though the consistency of MCC is apparent from the fact that it was the first metric to achieve a certain ranking across all datasets.

5.2 Probability Calibration

RQ2 asks *How do different feature selection metrics impact model calibration?* The features that are selected can influence how well it is theoretically possible to calibrate a model. For example, a model with two binary features can only output four possible values, and thus it is quite likely the model will be unable to output predicted probabilities that closely align with the true probability that the model’s prediction is correct or not.

Results show that RMSE easily produced the best results (Table 2), with a mean calibration score (CAL) of 0.166 and the best CAL score in 8 of the 11 datasets. Recall had the worst calibration score averaged across models and datasets, followed by precision, accuracy and F_1 .

5.3 Positive Class Predicted Rate

RQ3 asks *How do different feature selection metrics impact the predicted rates of models?* The predicted rate of models is in some respects related to model calibration, since

Table 1: Mean ranking for each metric and dataset. Lower is better, indicating that a metric, on average, yielded better results in terms of itself and the other metrics. Values range from 0 (selecting features for that metric always produced the best score in terms of itself and the other metrics) to 9 (the number of metrics). The best metric for each dataset is highlighted in green, while the worst is in red.

Dataset	Accuracy	AUC	F1	Kappa	MCC	MPAUC	Precision	RMSE	Recall
CTA-C	6.319	2.653	2.153	2.681	3.167	4.181	4.778	3.528	6.542
CTA-PF	5.403	2.222	4.903	4.625	3.986	2.208	6.069	3.736	2.847
VIDEO-ANU-C	2.958	4.069	3.583	3.403	4.194	3.806	6.556	5.250	2.181
VIDEO-HR-C	3.847	5.111	4.514	3.764	3.556	4.181	5.153	2.333	3.542
VIDEO-LBP-C	3.806	3.389	3.986	3.528	3.319	3.986	5.875	4.097	4.014
VIDEO-ANU-R	4.931	3.306	3.431	5.139	2.931	3.264	4.764	3.542	4.694
VIDEO-HR-R	2.000	4.389	4.111	4.333	3.583	4.722	5.319	3.306	4.236
VIDEO-LBP-R	3.833	2.361	6.458	2.528	4.056	3.069	4.597	3.306	5.792
EPM	3.222	5.319	3.208	2.458	3.125	2.694	6.056	3.556	6.361
MATH	5.556	3.569	1.583	3.333	2.222	2.653	6.472	4.056	6.556
PORTUGUESE	4.819	1.764	3.028	3.792	3.708	3.472	6.750	2.125	6.542
Mean	4.245	3.468	3.723	3.598	3.441	3.476	5.672	3.530	4.846
Std. dev.	1.282	1.172	1.327	0.862	0.573	0.776	0.781	0.843	1.605

Table 2: Mean calibration score of each metric for each dataset. Lower is better, where 0 indicates that predicted probabilities exactly matched the probability that that model’s predictions were correct. The best metric for each dataset is highlighted in green, while the worst is in red.

Dataset	Accuracy	AUC	F1	Kappa	MCC	MPAUC	Precision	RMSE	Recall
CTA-C	0.387	0.149	0.164	0.155	0.204	0.225	0.228	0.106	0.408
CTA-PF	0.337	0.282	0.269	0.268	0.269	0.260	0.403	0.252	0.261
VIDEO-ANU-C	0.271	0.281	0.281	0.263	0.278	0.261	0.320	0.257	0.271
VIDEO-HR-C	0.199	0.223	0.215	0.210	0.207	0.217	0.237	0.171	0.213
VIDEO-LBP-C	0.284	0.257	0.272	0.241	0.248	0.255	0.308	0.232	0.280
VIDEO-ANU-R	0.235	0.217	0.233	0.228	0.220	0.223	0.235	0.214	0.257
VIDEO-HR-R	0.199	0.194	0.209	0.199	0.198	0.205	0.217	0.173	0.202
VIDEO-LBP-R	0.211	0.193	0.239	0.201	0.219	0.214	0.249	0.199	0.249
EPM	0.067	0.147	0.069	0.060	0.066	0.071	0.099	0.063	0.184
MATH	0.086	0.132	0.138	0.141	0.137	0.130	0.083	0.091	0.137
PORTUGUESE	0.072	0.114	0.131	0.113	0.140	0.117	0.133	0.066	0.207
Mean	0.213	0.199	0.202	0.189	0.199	0.198	0.228	0.166	0.243
Std. dev.	0.106	0.059	0.068	0.065	0.063	0.063	0.096	0.073	0.070

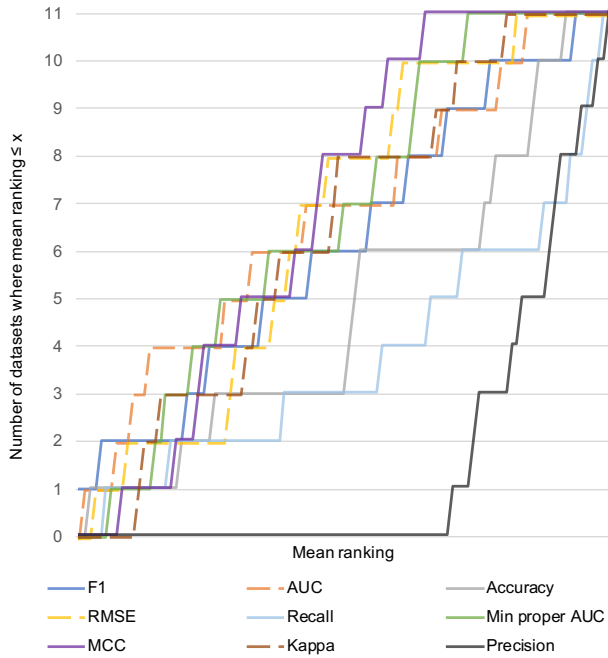


Figure 4: Step graph for mean rankings of metrics used for wrapper feature selection across all datasets. The left edge of the x axis indicates the best (lowest) ranking, while the right indicates the worst (highest). The y axis indicates the number of datasets that have mean rank $\leq x$.

a model that severely over- or under-predicts the positive class is unlikely to be well-calibrated (e.g., a model that always predicts 100% confidence for the positive class will have very poor calibration for any negative-class instances). Results reflect this calibration–predicted-rate relationship (Table 3), showing that selecting features based on recall resulted in the largest mean absolute difference between actual base rate and predicted rate (0.233), while RMSE was close to best (0.080). Selecting features based on accuracy (proportion correct) did not produce inaccurate predicted rates (mean absolute difference = 0.079), however, despite relatively poor model calibration.

For imbalanced datasets where classification is imperfect, accuracy can be inflated by over-predicting the majority class [10, 25]. However, Table 3 shows that selecting features based on accuracy did not have this effect, perhaps because we applied SMOTE to reduce the impact of class imbalance during training. Conversely, selecting features based on recall increased the positive class predicted rate for most datasets, since doing so can inflate recall regardless of the presence of class imbalance [10]. Similarly, selecting features based on precision often resulted in under-prediction of the positive class (10 out of 11 datasets).

5.4 Number of Features Selected

Selecting features based on precision yielded the fewest numbers on average (4.173), while selecting based on RMSE yielded the most (10.523). Selecting features based on AUC also yielded more features (10.006, on average) than other

metrics except RMSE.

These patterns are likely due to the fact that adding relatively unimportant features to a model will offer only marginal improvement, and may not be enough to shift predictions above or below the decision threshold. All of the metrics that require a decision threshold (accuracy, F_1 , kappa, MCC, MPAUC, precision, and recall) resulted in fewer features than the threshold-free metrics of AUC and RMSE. For example, adding a feature that applies to only a few instances may help push the probability decision for those few instances in the right direction, but may not change the binary decision for those instances and thus may not be selected when evaluating based on threshold-based metrics.

6. LIMITATIONS AND FUTURE WORK

There are a few limitations to the experiments in this paper. First, the datasets that we analyzed represent only a handful from among thousands of educational datasets that researchers and others have collected over the years. Our datasets are also quite diverse, measuring very different student characteristics. Thus, we have only a sparse sampling of the space of educational datasets, and datasets that vary notably from those reported on here could exhibit different trends. Future work is especially needed in this area to discover specific properties of datasets (e.g., number of features, type of features) that inform which metrics are likely to be successful for wrapper feature selection. Such analysis is only possible with a large enough number of datasets to enable statistical comparisons at the dataset level.

Second, the metrics we examined also only represent a subset of many possible. Many other metrics are closely related to those we studied (e.g., informedness, markedness, balanced accuracy), but may not exhibit exactly the same patterns. We selected a diverse mix of commonly reported metrics and some less-common metrics, all of which have been shown to be useful in previous research.

Third, we explored only four of the most prominent machine learning classifiers from among many possible options. We chose these classifiers because they are represented in many education-related research endeavors, but results for other classifiers may differ. Perhaps most importantly, deep neural networks are increasingly popular for educational data mining research [2, 28, 48, 47, 42], but were not considered here. Wrapper feature selection is perhaps less common for deep neural networks, given the high computational cost of model training, but correctness metrics often play a similar role in the model selection process for neural networks – for example, when deciding when to stop training a model. In future work we will explore issues of model selection for neural networks as well.

Fourth, averaging across the four classifiers is a limitation as well. While classifiers performed somewhat similarly, Figure 3 shows some exceptional cases. For example, kappa performed poorly with random forest, and precision performed well with logistic regression. As part of future work, we will explore classifier-based analysis of metrics in more depth, including statistical analyses (e.g., Friedman test) where we consider a large number of classifiers as judges that are ranking metrics.

Table 3: Mean predicted rate of the positive class for models with features selected based on each metric, for each dataset. Base rate indicates the actual proportion of the positive class in the dataset. The last row refers to the mean absolute difference between predicted rate and base rate across datasets. Green highlighting indicates the closest match to the true base rate, while red indicates the predicted rate furthest away in each row.

Dataset	Base rate	Accuracy	AUC	F1	Kappa	MCC	MPAUC	Precision	RMSE	Recall
CTA-C	0.068	0.060	0.179	0.170	0.169	0.215	0.250	0.149	0.111	0.695
CTA-PF	0.068	0.029	0.084	0.084	0.084	0.084	0.087	0.003	0.064	0.085
VIDEO-ANU-C	0.776	0.612	0.502	0.591	0.562	0.534	0.554	0.353	0.533	0.557
VIDEO-HR-C	0.776	0.669	0.688	0.669	0.724	0.705	0.703	0.681	0.753	0.663
VIDEO-LBP-C	0.776	0.631	0.526	0.586	0.617	0.563	0.548	0.385	0.588	0.607
VIDEO-ANU-R	0.733	0.610	0.637	0.567	0.594	0.616	0.611	0.581	0.657	0.568
VIDEO-HR-R	0.733	0.718	0.658	0.703	0.690	0.705	0.683	0.627	0.732	0.702
VIDEO-LBP-R	0.733	0.590	0.614	0.529	0.610	0.572	0.560	0.389	0.629	0.590
EPM	0.237	0.312	0.405	0.321	0.309	0.319	0.331	0.214	0.315	0.585
MATH	0.410	0.373	0.505	0.627	0.491	0.537	0.552	0.120	0.484	0.730
PORTUGUESE	0.425	0.437	0.524	0.609	0.509	0.573	0.532	0.085	0.473	0.840
Mean $ \Delta $		0.079	0.126	0.135	0.098	0.123	0.128	0.210	0.080	0.233

Table 4: Number of features in each dataset (N) and mean number of features selected by each metric. The highest number of selected features for each dataset is highlighted in light blue, while the lowest is highlighted in gray.

Dataset	N	Accuracy	AUC	F1	Kappa	MCC	MPAUC	Precision	RMSE	Recall
CTA-C	25	2.531	10.313	8.781	8.750	5.750	4.094	7.188	12.969	2.875
CTA-PF	60	10.469	35.219	25.125	24.625	26.125	28.156	1.000	28.094	27.969
VIDEO-ANU-C	42	3.656	5.031	3.875	4.500	4.531	4.625	3.219	5.438	4.031
VIDEO-HR-C	7	3.313	3.188	2.594	3.563	3.875	3.531	2.750	4.094	2.969
VIDEO-LBP-C	2304	3.563	6.344	4.031	5.750	5.781	4.844	2.000	7.656	3.625
VIDEO-ANU-R	42	4.281	6.063	5.063	5.594	4.906	4.813	3.875	7.688	4.281
VIDEO-HR-R	7	3.531	3.563	3.031	3.938	3.781	3.563	3.938	4.906	2.719
VIDEO-LBP-R	2304	8.344	12.656	6.500	9.125	9.500	9.469	6.500	16.781	4.750
EPM	38	6.375	6.344	6.219	7.125	6.688	5.813	7.156	7.406	1.000
MATH	43	7.000	8.719	6.438	8.656	7.781	7.781	4.313	8.250	1.375
PORTUGUESE	43	10.844	12.625	7.719	11.344	9.531	9.719	3.969	12.469	1.313
Mean	446.818	5.810	10.006	7.216	8.452	8.023	7.855	4.173	10.523	5.173

7. CONCLUSION

As the field of educational data mining develops, and machine learning becomes increasingly popular for modeling student outcomes, it is imperative to deeply understand each step of the process and the influence researchers' choices have on models. Our experiments offer insight into the large differences that can arise from machine learning design decisions, specifically for feature selection. We showed that selecting features based on some metrics is rarely advisable (especially precision), and that the choice of metric has impacts not only on correctness measures but on other important properties of the resulting models, including calibration and size (number of features).

We found that MCC produced the overall best results across the other metrics in terms of mean ranking as a measure of well-rounded correctness across metrics. MCC was not the best selection metric for all the datasets; in fact, it was the most effective only for 2 of the 11 datasets we analyzed in this study. However, it was more consistently well-ranked than the other metrics. On the other hand, RMSE produced the best-calibrated models, which can also be an important consideration for applying student models that might benefit from easily-adjustable decision thresholds.

Student models are the driving forces in adaptive learning software. Thus, enhancing them will lead to better software for students and teachers. The results of this project will enable researchers to more accurately build models which predict student outcomes by informing the correctness metrics relied upon for feature selection. In particular, we suggest utilizing metrics like MCC and RMSE (if calibration is desirable) to yield models with well-rounded accuracy across metrics. We suggest avoiding recall, precision, and accuracy, even though accuracy is the default setting in some machine learning software.

8. REFERENCES

- [1] R. Ade. Students performance prediction using hybrid classifier technique in incremental learning. *International Journal of Business Intelligence and Data Mining*, 15(2):173–189, Jan. 2019.
- [2] F. Ai, Y. Chen, Y. Guo, Y. Zhao, Z. Wang, and G. Fu. Concept-aware deep knowledge tracing and exercise recommendation in an online learning system. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)*, pages 240–245, 2019.
- [3] V. Aleven, B. McLaren, I. Roll, and K. Koedinger. Toward tutoring help seeking. In *International Conference on Intelligent Tutoring Systems*, pages 227–239. Springer, 2004.
- [4] H. Almayan and W. Al Mayyan. Improving accuracy of students' final grade prediction model using pso. In *2016 6th International Conference on Information Communication and Management (ICIM)*, pages 35–39. IEEE, 2016.
- [5] E. A. Amrieh, T. Hamtini, and I. Aljarah. Mining educational data to predict student's academic performance using ensemble methods. *International Journal of Database Theory and Application*, 9(8):119–136, 2016.
- [6] R. S. Baker, A. T. Corbett, K. R. Koedinger, and A. Z. Wagner. Off-task behavior in the cognitive tutor classroom: When students "game the system". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 383–390, New York, NY, USA, 2004. ACM.
- [7] R. S. Baker and P. S. Inventado. Educational data mining and learning analytics. In *Learning analytics*, pages 61–75. Springer, 2014.
- [8] N. Bosch, R. W. Crues, G. M. Henricks, M. Perry, L. Angrave, N. Shaik, S. Bhat, and C. J. Anderson. Modeling key differences in underrepresented students' interactions with an online STEM course. In *Proceedings of TechMindSociety '18*, pages 6:1–6:6, New York, NY, 2018. ACM.
- [9] N. Bosch and S. K. D'Mello. Automatic detection of mind wandering from video in the lab and in the classroom. *IEEE Transactions on Affective Computing*, in press.
- [10] N. Bosch and L. Paquette. Metrics for discrete student models: Chance levels, comparisons, and use cases. *Journal of Learning Analytics*, 5(2):86–104, 2018.
- [11] C. Burgos, M. L. Campanario, D. de la Peña, J. A. Lara, D. Lizcano, and M. A. Martínez. Data mining for modeling students' performance: A tutoring action plan to prevent academic dropout. *Computers & Electrical Engineering*, 66:541–556, 2018.
- [12] R. Caruana and A. Niculescu-Mizil. Data mining in metric space: An empirical analysis of supervised learning performance criteria. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 69–78, New York, NY, 2004. ACM.
- [13] P. Chaudhury, S. Mishra, H. K. Tripathy, and B. Kishore. Enhancing the capabilities of student result prediction system. In *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, pages 1–6, 2016.
- [14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2011.
- [15] D. Chicco and G. Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6, Jan. 2020.
- [16] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.
- [17] P. Cortez and A. M. G. Silva. Using data mining to predict secondary school student performance. In *Proceedings of 5th Annual Future Business Technology Conference*. EUROSIS-ETI, 2008.
- [18] T. Devasia, T. Vinushree, and V. Hegde. Prediction of students performance using educational data mining. In *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*, pages 91–95. IEEE, 2016.
- [19] P. Ekman and W. V. Friesen. *Facial action coding system: A technique for the measurement of facial movement*. Palo Alto, CA: Consulting Psychologists

- Press, 1978.
- [20] B. Guo, R. Zhang, G. Xu, C. Shi, and L. Yang. Predicting students performance in educational data mining. In *2015 International Symposium on Educational Technology (ISET)*, pages 125–128. IEEE, 2015.
 - [21] W. Hämmäläinen and M. Vinni. Classifiers for educational data mining. *Handbook of Educational Data Mining, Chapman & Hall/CRC Data Mining and Knowledge Discovery Series*, pages 57–71, 2011.
 - [22] M. Hussain, W. Zhu, W. Zhang, S. M. R. Abidi, and S. Ali. Using machine learning to predict student difficulties from learning session data. *Artificial Intelligence Review*, 52(1):381–407, 2019.
 - [23] M. Hussain, W. Zhu, W. Zhang, J. Ni, Z. U. Khan, and S. Hussain. Identifying beneficial sessions in an e-learning system using machine learning techniques. In *2018 IEEE Conference on Big Data and Analytics (ICBDA)*, pages 123–128. IEEE, 2018.
 - [24] D. Ifenthaler and C. Widanapathirana. Development and validation of a learning analytics framework: Two case studies using support vector machines. *Technology, Knowledge and Learning*, 19(1-2):221–240, 2014.
 - [25] L. A. Jeni, J. F. Cohn, and F. De la Torre. Facing imbalanced data—Recommendations for the use of performance metrics. In *Proceedings of the 5th International Conference on Affective Computing and Intelligent Interaction*, pages 245–251, Sept. 2013.
 - [26] M. Jovanovic, M. Vukicevic, M. Milovanovic, and M. Minovic. Using data mining on student behavior and cognitive style data for improving e-learning systems: a case study. *International Journal of Computational Intelligence Systems*, 5(3):597–610, 2012.
 - [27] A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and information systems*, 12(1):95–116, 2007.
 - [28] B.-H. Kim, E. Vizitei, and V. Ganapathi. GritNet: Student performance prediction with deep learning. In *Proceedings of the 11th International Conference on Educational Data Mining (EDM 2018)*, pages 625–629, 2018.
 - [29] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.
 - [30] B. W. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451, Oct. 1975.
 - [31] H. Monkaresi, N. Bosch, R. A. Calvo, and S. K. D’Mello. Automated detection of engagement using video-based estimation of facial expressions and heart rate. *IEEE Transactions on Affective Computing*, 8(1):15–28, 2016.
 - [32] H. Monkaresi, R. A. Calvo, and H. Yan. A machine learning approach to improve contactless heart rate monitoring using a webcam. *IEEE Journal of Biomedical and Health Informatics*, 18(4):1153–1160, July 2014.
 - [33] A. Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML 2004)*, ICML ’04, pages 78–85, New York, NY, 2004. ACM.
 - [34] R. Nilsson, J. M. Pena, J. Björkegren, and J. Tegnér. Evaluating feature selection for svms in high dimensions. In *European Conference on Machine Learning*, pages 719–726. Springer, 2006.
 - [35] Z. Papamitsiou and A. A. Economides. Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. *Journal of Educational Technology & Society*, 17(4):49–64, 2014.
 - [36] L. Paquette and R. S. Baker. Comparing machine learning to knowledge engineering for student behavior modeling: a case study in gaming the system. *Interactive Learning Environments*, 27(5-6):585–597, 2019.
 - [37] L. Paquette, N. Bosch, E. Mercier, J. Jung, S. Shehab, and Y. Tong. Matching data-driven models of group interactions to video analysis of collaborative problem solving on tablet computers. In J. Kay and R. Luckin, editors, *Proceedings of the 13th International Conference of the Learning Sciences (ICLS) 2018, Volume 1*, pages 312–319, London, UK, 2018. International Society of the Learning Sciences.
 - [38] S. Parodi, V. Pistoia, and M. Muselli. Not proper roc curves as new tool for the analysis of differentially expressed genes in microarray experiments. *BMC bioinformatics*, 9(1):410, 2008.
 - [39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, Nov. 2011.
 - [40] R. Pelánek. Metrics for evaluation of student models. *Journal of Educational Data Mining*, 7(2):1–19, 2015.
 - [41] S. Raschka. Mlxtend: Providing machine learning and data science utilities and extensions to python’s scientific computing stack. *The Journal of Open Source Software*, 3(24), Apr. 2018.
 - [42] J. M. Reilly and C. Dede. Exploring stealth assessment via deep learning in an open-ended virtual environment. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)*, pages 643–646, 2019.
 - [43] C. Romero, M.-I. López, J.-M. Luna, and S. Ventura. Predicting students’ final performance from participation in on-line discussion forums. *Computers & Education*, 68:458–472, 2013.
 - [44] T. Saito and M. Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3), 2015.
 - [45] F. Siraj and M. A. Abdoulha. Uncovering hidden information within university’s student enrollment data using data mining. In *2009 Third Asia International Conference on Modelling & Simulation*, pages 413–418. IEEE, 2009.
 - [46] N. Tasnim, M. K. Paul, and A. S. Sattar. Performance

analysis of different decision tree based methods for identifying drop out students. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–6. IEEE, 2019.

- [47] A. Tato, R. Nkambou, and A. Dufresne. Hybrid deep neural networks to predict socio-moral reasoning skills. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)*, pages 623–626, 2019.
- [48] T. Wang, F. Ma, and J. Gao. Deep hierarchical knowledge tracing. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)*, pages 671–674, 2019.
- [49] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svms. In *Advances in neural information processing systems*, pages 668–674, 2001.
- [50] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, June 2007.

APPENDIX

A. OVERALL CORRECTNESS AND COMPARISON TO PREVIOUS RESULTS

In this appendix we provide the correctness results obtained from our experiments, as a point of comparison to previous work and for comparison in future work. We report results where we selected features via MCC, since that metric had the best mean ranking in terms of other metrics (Table 1). We report only macro-level averaging, though results were similar for micro-level averaging (Figure 3). We also average results across all four classifiers, rather than selecting only the best classifier (and thus potentially introducing Type I error).

Table 5 shows the overall correctness metrics, with comparisons to previous work (where possible) noted by highlighted colors. For MATH and PORTUGUESE datasets we could not make direct comparisons because previous results did not perform median splitting to transform regression to binary classification. The feature selection criterion (which metric was used for selection) used by previous analysis is not clear in most cases. Hence, it is difficult to make close comparisons to previous models.

For the VIDEO-* datasets we compared AUC to [31]. In [31], AUCs reported were VIDEO-ANU-C: 0.635, VIDEO-HR-C: 0.544, VIDEO-LBP-C: 0.645, VIDEO-ANU-R: 0.666, VIDEO-HR-R: 0.590 and VIDEO-LBP-R: 0.644.

For the CTA-C dataset we compared AUC and kappa to [36]. However, the other models in this paper include feature-level fusion of both CTA-C and CTA-PF features, so they are not directly comparable to the CTA-PF features that we have. Reported values for CTA-C were AUC = 0.865 and kappa = 0.332.

For the EPM dataset we compared results to [22], which reports accuracy, F1, kappa, RMSE, precision and recall. However, accuracy, F1, precision and recall are reported for the random division and the alpha-investing feature selection methods and hence are not comparable to our results. The values (averaged across the reported models) were kappa = 0.443 and RMSE = 0.490, though the division of grades into two categories may have been based on a different split value than we utilized in this paper (the median), so comparisons should be made with that in mind.

Table 5: Our results for all metrics and datasets using MCC as the selection metric and macro-level averaging. Where previous results are known, green highlighting indicates that models we trained were better (more accurate) and red indicates that they were worse. Specific previous results are reported in the Appendix text.

Dataset	Accuracy	AUC	F1	Kappa	MCC	MPAUC	Precision	RMSE	Recall
CTA-C	0.819	0.874	0.368	0.295	0.363	0.796	0.242	0.361	0.770
CTA-PF	0.919	0.740	0.466	0.423	0.427	0.735	0.425	0.357	0.521
VIDEO-ANU-C	0.501	0.500	0.588	-0.004	-0.010	0.491	0.765	0.536	0.516
VIDEO-HR-C	0.654	0.565	0.762	0.102	0.107	0.554	0.801	0.483	0.738
VIDEO-LBP-C	0.534	0.511	0.633	0.006	0.002	0.500	0.773	0.518	0.570
VIDEO-ANU-R	0.558	0.552	0.636	0.035	0.039	0.521	0.747	0.514	0.585
VIDEO-HR-R	0.622	0.536	0.729	0.072	0.080	0.539	0.750	0.496	0.727
VIDEO-LBP-R	0.560	0.568	0.646	0.067	0.076	0.545	0.758	0.508	0.577
EPM	0.871	0.915	0.764	0.678	0.695	0.882	0.667	0.322	0.901
MATH	0.619	0.667	0.599	0.252	0.269	0.632	0.532	0.507	0.706
PORTUGUESE	0.656	0.722	0.659	0.331	0.362	0.675	0.571	0.491	0.797

Learning a Policy Primes Quality Control: Towards Evidence-Based Automation of Learning Engineering

Machi Shimmei

North Carolina State University
Raleigh, NC 27695
mshimme@ncsu.edu

Noboru Matsuda

North Carolina State University
Raleigh, NC 27695
noboru.matsuda@ncsu.edu

ABSTRACT

One of the most challenging issues for online courseware engineering is to maintain the quality of instructional components, such as written text, video, and assessments. Learning engineers would like to know how individual instructional components contributed to students' learning. However, it is a hard task because it requires significant expertise in learning science, learning technology, and subject matter pedagogy. To address this challenge, we propose an innovative application of reinforcement learning (RL) as an assessor of instructional components implemented in given online courseware. After students activities are converted into Markov decision process (MDP), a collection of actions (each corresponds to an instructional component) suggested as a policy is analyzed. As a consequence, the usefulness of individual actions with regards to achieving ideal learning outcomes will be suggested. The proposed RL application is invented for human-in-the-loop learning engineering method called RAFINE. In the RAFINE framework, a machine generates a list of the least contributing instructional components on the given online courseware by interpreting the whole policy. The courseware developers modify those suggested components. As a proof of concept, this paper describes an evaluation study where online learning was simulated on hypothetical online courseware. The results showed that over 90% of ineffective instructional components were correctly identified as ineffective on average.

Keywords

Automated Learning Engineering, Evidence-based learning Engineering, Iterative Courseware development, Reinforcement Learning

1. INTRODUCTION

With the rapidly growing popularity of online courses, there has been a heavy demand for practical learning engineering methods for designing effective online courseware [18]. Even though there are known design principles that provide theoretical insights into designing effective online courses [e.g., 5, 6, 10], such principle-

based approaches still require iterative engineering for practical courseware development at scale [8].

One of the challenges in the principle-based iterative learning engineering is to identify issues with the courseware. After an initial version of courseware is used by students, instructors (or learning engineers) analyze the interaction and learning outcome data to improve the quality of the courseware. However, interpreting those data to determine actual refinement plans is extremely challenging and requires significant expertise in learning science, learning technology, and the subject matter pedagogy [5, 20]. The commonly used analytic techniques are the learning curve analysis [14] and the assessment items analysis [16]. However, these techniques only apply to assessment items while other types of instructional components such as video clips and written texts must also be included in the analysis.

There is therefore a gap between an ideal learning-engineering model to *efficiently* build *effective* online courseware and the actual technology infrastructure available for building online courseware. To fill this gap, evidence-based learning engineering method that identifies deficits of the given online courseware is needed.

Our solution is an innovative application of the reinforcement learning (RL) technique that we call RAFINE (Reinforcement learning Application For INcremental courseware Engineering). RAFINE identifies instructional components that have relatively less contribution to students' learning in the given courseware. RAFINE is a building block for the evidence-based, human-in-the-loop, iterative learning engineering method that we call the RAFINE method. Figure 1 shows how a human and a machine collaborate to iteratively improve the quality of courseware in the RAFINE method.

Given a record of individual students' learning trajectory logs on particular online courseware, RAFINE first converts learning trajectories into a state transition graph. Here, states represent students' intermediate learning status and the transition is caused by taking an instructional component (i.e., watching a video). All students' state transition graphs are then consolidated into a single Markov decision process (MDP) by merging the same states. RAFINE then applies a variant version of value iteration technique commonly used for RL to compute a *converse policy* that represents the least optimal instructional components to be taken at any given moment to achieve students' ideal learning goals. The entire policy actions will be then analyzed to identify instructional components that have relatively less contributions to students' learning. The list of detected less-effective components is provided to instructors as a *recommendation* for courseware refinement. Any type of

Machi Shimmei and Noboru Matsuda "Learning a Policy Primes Quality Control: Towards Evidence-Based Automation of Learning Engineering" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 224 - 232

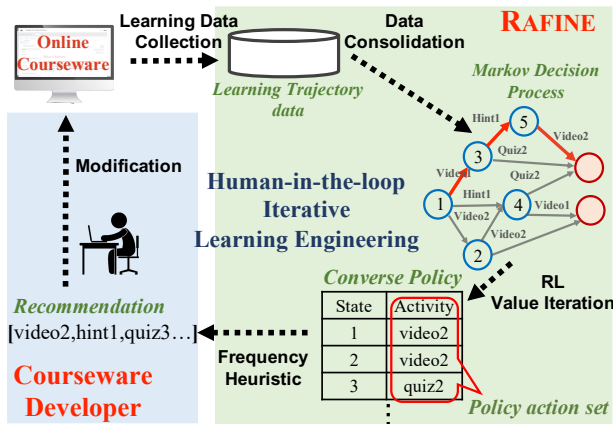


Figure 1 : Overview of the RAFINE method

instructional components such as lecture videos, assessment quizzes and hints can be analyzed by the RAFINE method.

As a proof of concept, the goal of the current paper is to conduct a study to evaluate the validity of implementation of the RAFINE method. We conducted a simulation study where learning log data were generated by simulating learning activities in mock online courseware. The result showed that the RAFINE method correctly identified over 90% of ineffective instructional components in the given mock courseware.

The primary contributions of the current work are essentially theoretical yet have the potential for practical use:

- (1) We proposed RAFINE as a building block for evidence-based, human-in-the-loop, iterative online courseware learning engineering method. RAFINE analyzes complicated learning trajectory data and suggests deficits of the courseware by evaluating a broader range of instructional components. The evaluation study showed the theoretical effectiveness of RAFINE.
- (2) We innovated a technique to interpret a policy induced by reinforcement learning as a whole to detect a relative weakness among available actions (which correspond to instructional components) with regard to achieving ideal goals.

The rest of the paper is structured as follows. Section 2 discusses related work on evidence-based learning engineering and the applications of reinforcement learning for education. Section 3 technically elaborates how RL works on learning trajectory data and describes how RAFINE interprets the converse policy. Section 4 introduces research questions. Then, section 5 explains how simulation data were created for the evaluation study. Section 6 reports results of the evaluation study along with the research questions, and section 7 discusses the results and limitations. Finally, section 8 concludes that RAFINE serves as a building block for the evidence-based iterative learning engineering method.

2. RELATED WORK

2.1 Evidence-based Learning Engineering

The process of learning engineering includes cognitive task analysis, designing and delivering instructional components, measuring students' understanding, and evaluating the courseware design [6]. Quickly cycling through these tasks is a key for successful iterative improvement of online courseware. Since each of these tasks is considerably labor intensive, the automatization of the engineering process is an important research agenda.

One of the most actively studied areas of learning engineering is student modeling [14]. Both a representation (to understand what must be modeled) and a recognition (to understand how to gauge students' competency with the proposed representation) are important research topics. Cen *et al.* [3] proposed Learning Factors Analysis (LFA) for semi-automated evaluation and improvement of knowledge component (KC) models that represent a set of latent skills and knowledge that students are supposed to learn [12]. LFA performs a combinational search for a KC model that best fits students' learning data across existing KC models. Although LFA requires "seed" KC models, some more recent works reported automatic discovery of KC models from students' learning data [9, 13].

Another actively studied area of learning engineering is an automated question generation. Du and Cardie [7] proposed a method for automatically generating questions. The method identifies the question-worthy sentences from a passage using a hierarchical neural model with a sentence-level sequence tagging. Mazidi and Tarau [15] introduced a method to classify sentences based on what the sentence is communicating as a basis for generating questions. The type of syntactic and semantic constituents of sentences and their arrangement were analyzed in the study.

Automation of assessment grading is also an important part of learning engineering in particular for online courses to be scaled. Zhang *et al.* [25] tackled the task of Automatic Short Answer Grading (ASAG). Short answer questions ask students to answer in natural language with the length of one phrase to one paragraph. The authors compared Deep Belief Networks (DBN) against five machine learning techniques (Naïve Bayes, Logistic Regression, Decision Tree, Artificial Neural Network, and Support Vector Machine) for automatically grading short answer questions.

Yet another essential part of practical learning engineering is to identify deficits of courseware content to be revised. RAFINE focuses on this aspect of the learning engineering. As far as the authors are aware, there have been very few studies in this line of research. Bodily *et al.* [2] mentioned the lack of efforts to use learning analytics as a basis for redesigning an online course. The authors proposed the RISE framework for redesigning instructional components in online courseware. This framework provides a metric that combines the usage of instructional components and students' grades to identify instructional components that are good candidates for improvement efforts. The goal of the RAFINE method is also to detect instructional components that were not very useful for learning. Unlike the RISE framework, however, RAFINE evaluates a potential contribution of implemented instructional components to students' learning and provides a list of instructional components as a recommendation for refinement.

2.2 RL for Decision Making in Education

Reinforcement Learning (RL) has been used in education applications in particular to compute optimal pedagogical strategies for adaptive tutoring.

Shen and Chi [19] applied RL to induce the policy on whether the intelligent tutoring system should propose worked example (WE) or problem solving (PS) to students for the next activity. The authors induced policies using two different rewards: Immediate and Delayed. The policy was computed based on the number of problems solved, the average time taken, the difficulty of the problem, and students' performance on the past PS. The result

showed that immediate policies give more WE while delayed policies give more PS.

Rafferty *et al.* [17] used the partially observable Markov decision process (POMDP) framework to formulate the process of teaching. The authors applied RL to induce optimal teaching actions such as a quiz or an example to minimize the amount of time spent on learning materials. They found that students who learned with RL induced teaching actions spent less time than students who did not.

Iglesias *et al.* [11] applied RL to induce a policy on teaching decisions such as which topic students should do next and which task students should do on the topic. They conducted an evaluation study in a database design course for undergrad students in Computer Science. In the evaluation study, they found that students with the machine generated policy spent less time on the adaptive and intelligent educational system, but they could not find a significant difference in the students' final level of knowledge.

In addition to inducing the sequence of instructional components, RL has been applied to induce dialog moves or narrative events. Chi *et al.* [4] applied RL to induce pedagogical policies that decide whether the tutor should ask students to justify the answer, tell the next step directly, or elicit the next step information from a student in a dialogue-based tutor. Tetreault and Litman [21] estimated the reliability of a policy derived from a spoken dialog tutoring system. Wang *et al.* [22] applied deep RL for interactive narrative generators that tailor each player's story in an educational game. The authors prepared several events of a story and induced policies on how event sequences should unfold based on player interaction logs.

The applications of reinforcement learning to induce pedagogical strategies are widely studied in various subjects from middle school math to college-level database design, and in various kinds of tutoring systems such as task-based, dialog-based, and game-based systems. The effects of educational RL policy have been tested both with real and simulated data. Some showed positive effects of the policy while others did not.

What makes our study different from these studies is the way we use the induced policy. In the previous studies mentioned above, the induced policy is directly used to provide an *optimal action at each learning status*. On the other hand, RAFINE does not use the policy to make a decision on which instructional component students should take next. Instead, RAFINE interprets the induced policy as a whole to identify instructional components that have relatively less contribution to learning. More specifically, RAFINE focuses on how often each instructional component is suggested by a policy (we call this the *frequency heuristic* as described in section 3.5). In RAFINE, the induced policy is not utilized as an educational strategy for students, but an analysis of the policy is used for courseware developers to improve courseware.

3. TECHNICAL DETAILS OF RAFINE

3.1 Overview of the RAFINE Method

In the RAFINE method, an initial version of the online courseware is used by students and their activities are logged. These activity data consist of standard clickstream data including students' responses for formative assessments and their correctness. We call these activity data the *learning trajectory data*.

The right side of Figure 1 shows how learning trajectory data are processed. The learning trajectory data from all students are first consolidated into a single state transition graph called *learning trajectory graph* (LTG). The LTG is a Markov decision process

(MDP) where states represent students' intermediate learning status and actions represent instructional components taken. LTG is annotated with predefined *rewards* that represent quantitative benefits of the learning activity that causes transition from one state to another in the LTG. Finally, a value iteration technique is applied to compute a *converse policy* that shows the worst action to be taken at each state to achieve the expected learning outcome (represented as a table in Figure 1). As a consequence, a collection of actions suggested by a converse policy corresponds to a set of instructional components that have the least likelihood at each state to contribute to the ideal learning outcome. We call this collection of actions the *policy action set*.

To create a recommendation for refinement based on the induced policy, RAFINE *interprets the policy as a whole*. That is, all actions in a policy action set is analyzed. Note that in most cases, the number of states in the LTG gets larger than the number of instructional components available on the given online courseware. This implies that all instructional components are likely to be included in a policy action set. The relative effectiveness of individual instructional components is therefore analyzed based on the frequency. We call this heuristic the *frequency heuristic*, which is detailed in section 3.5.

Given the recommendation for refinement, courseware developers revise the courseware. The RAFINE method can be iteratively applied to the revised courseware by collecting a new batch of learning trajectory data to further improve the courseware.

3.2 Model Representation

The unit of analysis of the RAFINE method is an instructional component implemented in the online courseware. Instructional components include video, quiz, hint, written paragraph or any other components used in the courseware. We assume a presence of a skill model that contains a set of skills each representing a piece of knowledge that students must learn (aka, knowledge component), and each instructional component is tagged with a single skill. Under this assumption, RAFINE will be applied for each skill separately. This constraint is rooted in our design decision for a state representation described later that involves a measure of proficiency per skill.

Let LT_i^ϕ be a given learning trajectory for student i regarding skill ϕ . Let a_i^T be an instructional component taken (e.g., watching a video or answering a quiz) by student i at time T . A learning trajectory for student i on skill ϕ , LT_i^ϕ , is represented with a_i^T as follows:

$$LT_i^\phi = \{a_i^1, \dots, a_i^{n_i^\phi} \mid a_i^T \in \Phi^\phi, T = 1, \dots, n_i^\phi\}.$$

LT_i^ϕ : learning trajectory for student i regarding skill ϕ

a_i^T : an instructional component taken by student i at time T

Φ^ϕ : a set of instructional components regarding skill ϕ

n_i^ϕ : number of activities taken by student i regarding skill ϕ

To make the explanations simple, without a loss of generality, let's assume that there is only one skill ϕ in our target online courseware (recall that RAFINE will be applied for individual skills separately). We therefore eliminate the skill index from Φ^ϕ and LT_i^ϕ in the following descriptions unless otherwise desired.

All learning trajectories LT_i for all students i in the given log data are consolidated into a single learning trajectory graph (LTG), which is an MDP. In the LTG, states represent learning status and

edges represent learning activities taken that caused a change in learning status. To consolidate individual students' learning trajectories into a single LTG, each student's learning trajectory LT_i is first converted into a *learning trajectory path*. This is done by chronologically traversing a learning trajectory LT_i while creating states that represent intermediate learning status.

A learning status consists of a pair of *action history* and *mastery level*; $\langle \mathbf{ah}_{i,T}, p_{i,T}(\phi) \rangle$. Action history $\mathbf{ah}_{i,T}$ is a binary vector $\langle ah_i^1, \dots, ah_i^K \rangle$ where ah_i^m shows whether student i has taken the m -th instructional component in Φ^ϕ by time T (assuming the instructional components are ordered and $|\Phi^\phi| = K$). Mastery level $p_{i,T}(\phi)$ is a scalar value showing a predicted probability of student i applying skill ϕ at time T correctly. The value of mastery level is rounded to the nearest multiple of 0.05 (e.g., 0.12 becomes 0.10) to reduce the number of states in the LTG (which will be otherwise intractable).

Mastery level is computed based on the history of learning activities. An underlying assumption is that commitment to a particular type of learning activity would increase the mastery level by a specific amount. There are several known techniques available to achieve this goal including Bayesian models and regression models. As long as mastery level is monotonically updated, any student-modeling technique would work for the RAFFINE method.

While traversing the learning trajectory, $\mathbf{ah}_{i,T}$ and $p_{i,T}(\phi)$ are updated accordingly. For example, assume there are six instructional components; Video1, Video2, Quiz1, Quiz2, Hint1, and Hint2. A state $s \langle 101000, 0.4 \rangle$ indicates that a student had watched Video1 and took Quiz1 before reaching the state s . It also indicates that a predicted mastery level for skill ϕ at the time of arriving at the state s was 0.4. Assume that the student answered Quiz1 incorrectly to reach the state s . Now, the student needed to review Hint1, which caused a transition from s to s' where s' is $\langle 101010, 0.45 \rangle$ with an assumption that reviewing a hint increased the master level by 0.05.

A learning trajectory path is a linear graph. It might have a loop back to the same state when a certain instructional component was taken more than once with the increase of mastery level less than 0.05. As a side note, moving between pages in the courseware is not encoded in the LTG, because it is not considered as a learning activity.

All individual students' learning trajectory paths are then aggregated into an LTG by merging the same states. As a consequence, the states in an LTG generally have multiple incoming and outgoing edges. Note that in an LTG, student and time (i.e., the parameters i and T in an individual student's learning trajectory path) are abstracted. Therefore, in the following explanations, a tuple representing a state is denoted as $\langle \mathbf{ah}, p(\phi) \rangle$.

In an LTG, the states where the value of the mastery level, $p(\phi)$, is greater than a pre-defined threshold (which is usually 0.85) are called *terminal states*—meaning that students became proficient in applying skill ϕ . All outgoing edges at terminal states are discarded.

3.3 Reward

A reward value of a particular state depends on the mastery level, $p(\phi)$, both at the current and successor states. As an example, consider two students who landed on the same state s , but then took different learning activities. One student reached a successor state by answering an assessment quiz incorrectly (i.e., $p(\phi)$ was not increased) whereas the other student watched a video (i.e., $p(\phi)$ was

increased). In our model, a reward for state s where the student took a learning activity a to reach a successor state s' is defined as:

$$R(s, a, s') = \begin{cases} -0.14 & (ml(s) = ml(s') < 0.85) \\ -0.05 & (ml(s) < ml(s') < 0.85) \\ 0.95 & (0.85 \leq ml(s')) \end{cases}$$

In the equations above, $ml(s)$ returns the mastery level at the state s . A reward at the state s becomes the greatest when the successor state is a terminal state. Otherwise, the rewards are set to be small negative values to reflect students' time commitment while computing a policy as shown in the next section. We assume that the mastery level grows monotonic, i.e., students never unlearn. Therefore, a reward where $ml(s) > ml(s')$ is undefined.

3.4 Converse Policy

Given the reward function R , a value function for state s under a policy π is defined as follows, where \mathbf{S} is a set of all states in a given LTG:

$$V^\pi(s) = \sum_{s' \in \mathbf{S}} T(s, \pi(s), s') \{R(s, \pi(s), s') + \gamma V^\pi(s')\}$$

In the current implementation, the discount factor γ is arbitrarily set to be 0.9. A transition model $T(s, a, s')$ is derived from the collected learning trajectory data of actual students as the probability of students reaching state s' when they took a learning activity a at state s .

In general, a policy suggests an action to be taken in a certain state to maximize the value function [23]. However, considering the purpose of RAFFINE, we need to know which instructional components should not be taken—i.e., we need to know which action has the least expected reward. Therefore, through the value iteration, the value function is updated as follows where $A(s)$ shows a set of actions available at state s (i.e., instructional components taken by students at state s):

$$V(s) \leftarrow \min_{a \in A(s)} \sum_{s' \in \mathbf{S}} T(s, a, s') \{R(s, a, s') + \gamma V(s')\}$$

After the value function is converged, the action that minimizes the value function for state s is identified. We shall call this policy the *converse policy*:

$$\pi(s) = \operatorname{argmin}_{a \in A(s)} \sum_{s' \in \mathbf{S}} T(s, a, s') \{R(s, a, s') + \gamma V^\pi(s')\}$$

3.5 Frequency Heuristic

Because of the binary vector and mastery level in the state representation, the number of states in any given learning trajectory graph (LTG) is many times more than the number of available actions (i.e., instructional components). Hence it is often the case that each of the instructional components in the courseware is selected as a policy action many times. Therefore, whether the instructional component has been selected as an action is not a sufficient criterion to decide the component should be included in a recommendation for refinement.

To create a recommendation from the induced converse policy, RAFFINE interprets the collection of policy actions over all states in the LTG based on a frequency—*actions that frequently appear in the converse policy action set* will be included in a recommendation as culprit for poor performance. We call this heuristic the *frequency heuristic*. The frequency heuristic is based on the hypothesis that relatively ineffective instructional components tend to appear in a

policy action set of a given converse policy more frequently than effective ones. We will verify this hypothesis by comparing the mean normalized frequency of ineffective and effective components in the policy action set.

The empirical question is then “how frequent is frequent?” We examined two frequency cut-offs through an evaluation study as described in the results section.

4. RESEARCH QUESTIONS

Our central question is whether RAFINE can suggest deficits of the courseware based on the students’ learning trajectory data. To address this question, we divide it into the following two research questions:

RQ1: How robust is the converse policy as a detector for relatively ineffective instructional components against different conditions of learning data?

RQ2: How accurately does the frequency heuristic compose a recommendation for courseware refinement?

To answer these questions, and also as a proof of concept for RAFINE, we conducted an evaluation study as described in the next section.

5. EVALUATION STUDY

For a rigorous evaluation of the RAFINE method, it is necessary to conduct a study with learning data collected from students working on actual online courseware. As mentioned earlier, the online courseware must be structured with a skill model tagged to individual instructional components to apply RAFINE. To the authors’ knowledge, however, no such online courseware is available at this moment and building RAFINE compatible online courseware requires a considerable amount of time. Therefore, we conducted a simulation study as a proof of concept towards an evaluation with actual students. The current evaluation study uses hypothetical learning trajectories in mock online courseware. The results from the current simulation study justify future efforts of building an RAFINE compatible online courseware or tagging a KC model to individual instructional components in existing courseware.

In this evaluation study, we address the research questions mentioned in section 4. We created mock online courseware where there was only one skill involved. As described above, when there were multiple skills involved, RAFINE had to be applied separately to each skill. Therefore, this assumption does not harm the generality of the study.

All instructional components in mock online courseware were tagged as either *effective* or *ineffective*. In the current simulation study, we included three types of instructional components: (1) videos, (2) formative assessments (aka quizzes), and (3) hint messages associated with formative assessments. Learning trajectories were generated by simulating students’ learning activities. For the sake of explanation, we use a phrase ‘simulated students’ to refer to hypothetical students in the simulation.

In the real world, the growth of mastery level depends on the learning activities actually taken and students’ latent traits of learning. In the current simulation, the mastery level shows a probability of answering a quiz correctly and the simulated students’ performance on a quiz was determined by the mastery level. The growth of the mastery level, $p_{i,T}$, was simulated using a logistic regression model as shown below:

Table 1: The means μ and standard deviations σ used for the simulation study to model the growth of mastery level.

(a)

The value of (μ_1, σ_1) where

$$\delta_1(c, e(a_{i,T-1})) \sim \max(0, \mathcal{N}(\mu_1, \sigma_1^2))$$

	Contrast: c			
Effectiveness: $e(a_{i,T-1})$	Large		Moderate	Small
Effective	0.5	0.01	0.4	0.05
Ineffective	-0.1	0.01	0.0	0.05

(b)

The value of (μ_2, σ_2) where

$$\delta_2(\text{rspns}(a_{i,T-1})) \sim \max(0, \mathcal{N}(\mu_2, \sigma_2^2)), \text{ or } 0 \text{ if } a_{i,T-1} \text{ was not a quiz.}$$

$\text{rspns}(a_{i,T-1} \in \text{quiz})$	μ_2	σ_2
Correct	0.05	0.01
Incorrect	0.03	0.01

Table 2: The means and standard deviations used for computing the initial logit $Z_{i,0}$.

The value of (μ_0, σ_0) where $Z_{i,0} \sim \max(0, \mathcal{N}(\mu_0, \sigma_0^2))$

	Contrast: c		
	Large	Moderate	Small
μ_0, σ_0	-0.95 0.01	-0.95 0.10	-0.95 0.20

$$p_{i,T} = \left[\frac{1}{1 + e^{-Z_{i,T}}} \right]$$

$$Z_{i,T} = Z_{i,T-1} + \delta_1(c, e(a_{i,T-1})) + \delta_2(\text{rspns}(a_{i,T-1}))$$

The $[X]$ operator is to round the value X to the nearest multiple of 0.05 and $a_{i,T-1}$ is an instructional component that a simulated student i took at time $T-1$.

Logit $Z_{i,T}$ was directly increased with $\delta_1(c, e(a_{i,T-1})) + \delta_2(\text{rspns}(a_{i,T-1}))$. δ_1 and δ_2 model learning gain obtained by taking an action $a_{i,T-1}$. $\delta_1(c, e(a_{i,T-1}))$ is a rectified random variable that follows a normal distribution with mean μ_1 and standard deviations σ_1 , i.e., $\max(0, \mathcal{N}(\mu_1, \sigma_1^2))$. μ_1 and σ_1 are given a priori based on c and $e(a_{i,T-1})$.

c and $e(a_{i,T-1})$ that represent *contrast* and *effectiveness* respectively were the parameters controlled to create several online learning scenarios for research question RQ1. We controlled the difference in impact on students’ learning (i.e., mastery level) between effective and ineffective instructional components using two parameters: (i) c that represents the *contrast* in the increase of logit between effective and ineffective instructional elements—large vs. moderate vs. small, and (ii) $e(a_{i,T-1})$ that represents the *effectiveness* of the instructional element $a_{i,T-1}$ —effective vs. ineffective.

The fundamental assumptions were that (1) the larger the contrast, the larger the differences of μ_1 when effective vs. ineffective instructional components were taken, and (2) the larger the contrast, the smaller the σ_1 was. For example, if c = “large”, (μ_1, σ_1) was (0.5, 0.01) vs. (-0.1, 0.01) for $e(a_{i,T-1})$ = “effective” vs. “ineffective.” However, they were (0.3, 0.1) vs. (0.1, 0.1) if c = “small.” Table 1(a) shows μ_1 and σ_1 for different *contrast* and *effectiveness*.

$\delta_2(\text{rspns}(a_{i,T-1}))$ is also a rectified random variable that follows a normal distribution with mean μ_2 and standard deviations σ_2 . The variable δ_2 was set to be zero if $a_{i,T-1}$ was not a quiz. Otherwise, μ_2 and σ_2 were determined a priori based on a student’s response

rate, $rsps = \text{correct}/\text{incorrect}$. We assume that when a student was able to answer the quiz correctly, $\text{logit } Z_{i,T}$ increases more than when the student was not able to answer it. Table 1(b) shows μ_2 and σ_2 for correct and incorrect responses respectively.

Student's initial $\text{logit } Z_{i,0}$ also followed a rectified normal distribution with μ_0 and σ_0^2 . These were given a priori based on the contrast parameter, c , as shown in Table 2.

In addition to three learning scenarios with different *contrasts*, we also created three versions of mock online courseware with different *qualities*. The quality of courseware was operationalized as the ratio of a number of effective to ineffective instructional components in the courseware. Three types of qualities are implemented in this study: High, Medium and Low. The higher the quality, the larger the proportion of effective instructional components. In the simulation study, each page in the mock online courseware included 3 lecture videos, 3 quizzes, and 3 hint messages each associated with a quiz. The low, medium, and high-quality courseware included 80-90%, 50-60%, and 10-20% ineffective instructional components.

Two instances of mock courseware (with a different number of pages) were created for each level of quality. Those six instances of courseware were crossed with three levels of contrast, resulting in 18 different simulated-learning scenarios. In each scenario, simulated students took a total of 10 to 30 instructional components.

Learning trajectories of students were randomly generated as follows. At first, for each simulated student, the number of instructional components to be taken was randomly decided. Either a video or a quiz was then randomly selected as the first learning activity. If it was a quiz, the student might show a hint before trying to answer the quiz at 0.05 probability. When the student answered a quiz, the correctness of the quiz response was determined randomly using the mastery level as the probability distribution. When the response was incorrect, either requesting a hint or retaking the same quiz (as a next instructional component) was randomly determined based on the probability distribution reported in [1]. Let quiz_x be a quiz with an ID x that student answered incorrectly. The probability distribution is as follows: (i) Try quiz_x at 0.78 probability, (ii) show hint_x at 0.20, (iii) give up and move to different quiz or video at 0.02 (these two are randomly selected). The same distribution is applied when the student showed hint_x . This process was repeated for the number of instructional components to be taken. Simulated students were able to retake the same instructional components.

For each of 18 learning scenarios, 100 course offerings were created each with 1,000 simulated students. In other words, this simulation study modeled a large-scale field trial as if 1800 instances of online course offerings were tested each with 1,000 student participants.

For each course-offering simulation, the learning trajectory data were converted into a learning trajectory graph (LTG). As a consequence, 1,800 instances of LTGs were generated. The manipulation of logit described above was used to estimate mastery level in LTG. For each of the 1,800 LTGs, the value iteration technique was applied to compute a converse policy. From each converse policy, the frequency heuristic was applied to generate a recommendation for refinement for a corresponding instance of online courseware.

6. RESULTS

6.1 Overview of the Data

To verify the feasibility of the simulation data, we computed a correlation between the ratio of effective to ineffective instructional components taken by a student and the final mastery level. The data showed a strong positive correlation, $r = 0.70$, $t(1799998) = 1314.56$, $p < 0.001$, suggesting that the final mastery level was significantly higher when simulated students took relatively more effective instructional components than ineffective ones.

6.2 Converse Policy-based Recommendation

6.2.1 Frequency Heuristic

The hypothesis under the frequency heuristic is that relatively ineffective instructional components tend to appear in a policy action set of a given converse policy more frequently than effective ones. To verify this hypothesis, we first compare the normalized frequency of ineffective components in the policy action set with that of effective ones. We also answer RQ1: How robust is the converse policy as a detector for ineffective instructional components against different conditions of learning data?

The frequency of an instructional component i selected as a converse policy action was normalized as follows. Let π be a converse policy and S be the set of states in the LTG. The normalized frequency of an instructional component i is calculated by the following equation.

$$\text{Normalized Frequency}(i) = NF(i) = \frac{|S^\pi(i)|}{|S^A(i)|}$$

$S^\pi(i) = \{s | \pi(s) = i\}$: A set of states in the LTG where i is the converse policy action.

$S^A(i) = \{s | i \in A_s\}$ (where A_s is a set of actions available from state s): A set of states where the instructional component i was taken.

$|X|$: Number of elements in X

Also notice that $S^\pi(i) \subset S^A(i)$.

We then tested if there was a significant difference in the mean normalized frequencies between effective and ineffective instructional components. Table 3 shows the mean normalized frequencies of ineffective and effective instructional components and those standard deviations. The effect size is a ratio of the difference between two means to the standard deviation. Table 3 suggests that regardless of the quality and contrast, ineffective instructional components were selected as a converse policy action notably many times more than effective ones. The differences were all statistically significant using t-test ($p < 0.01$). The data also suggest that the difference in the frequencies between ineffective and effective components becomes the smallest (as indicated by the smallest effect size) when contrast is small and quality is high, as we expected.

These results support the hypothesis that *relatively ineffective instructional components tend to appear in a converse policy action set more frequently than effective instructional components. It is also shown that the converse policy is robust enough to discriminate the effectiveness of the instructional component regardless of the quality (operationalized as the ratio of effective vs. ineffective components) and the contrasts (operationalized as the difference in the growth of logit between effective and ineffective components).*

Table 3: Comparison of the mean normalized frequency between ineffective (Inef.) and effective (Ef.) instructional components. A number in the parentheses shows an effect size.

Quality	Contrast					
	Large		Moderate		Small	
	Inef.	Ef.	Inef.	Ef.	Inef.	Ef.
High	0.7±0.2 (4.0)	0.2±0.1	0.7±0.1 (5.7)	0.1±0.1	0.5±0.1 (3.1)	0.2±0.1
Med.	0.4±0.1 (7.9)	0.1±0.05	0.4±0.1 (8.5)	0.1±0.04	0.4±0.1 (3.6)	0.2±0.1
Low	0.4±0.1 (9.2)	0.04±0.04	0.4±0.1 (10.0)	0.04±0.03	0.4±0.1 (4.5)	0.1±0.1

6.2.2 Accuracy of recommendation

We next evaluate the precision and recall scores of recommendations created by frequency heuristic to answer RQ2: How accurately does the frequency heuristic compose a recommendation?

To compose a recommendation, we need to define a cut-off value. As a reminder, those instructional components whose normalized frequency is more than a pre-defined cut-off are labeled as “ineffective” and included in the recommendation. What the cut-off value should be is an empirical call.

In the current study, we compared two cut-off values using mean (M) and standard deviation (SD) of the normalized frequency: M+SD vs. M-SD. To evaluate the accuracy of recommendation, we computed Precision and Recall as follows:

$$Precision = \frac{|\Phi_{ineff}^R|}{|\Phi^R|}$$

$$Recall = \frac{|\Phi_{ineff}^R|}{|\Phi_{ineff}|}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$|\Phi_{ineff}^R|$: Number of *ineffective* instructional components included in a recommendation

$|\Phi^R|$: Number of total instructional components included in a recommendation

$|\Phi_{ineff}|$: Number of *ineffective* instructional components in courseware

We investigated how precision and recall scores vary depending on the cut-off and the condition of the learning data (contrast, quality). Figure 2 shows precision and recall scores comparing M-SD and M+SD cut-offs for each quality of the courseware. For each data point, three levels of contrasts are aggregated, because there was no notable difference among them. The figure show that when the quality of courseware is low to medium, the M-SD cut-off had better recall and precision scores than M+SD. F1 score for M-SD was 0.99 and 0.92 for low and medium qualities respectively. On the other hand, when the quality is high, the M+SD cut-off outperformed M-SD. F1 scores of M+SD for high quality courseware was 0.88.

In sum, the frequency heuristic adequately works to determine which instructional components must be taken into a recommendation for courseware refinement. In the current simulation study, over 90% of ineffective instructional components

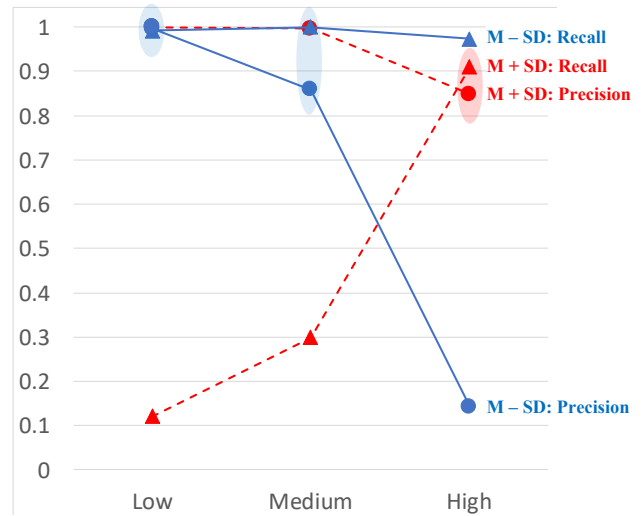


Figure 2 : Precision and recall of a recommendation. The X-axis represents the quality of the courseware. Red dashed lines show results from M+SD and blue solid lines show M-SD.

were correctly taken into a recommendation when an appropriate cut-off was used based on the maturity of the courseware. When the courseware is newly built (which is usually in a low to medium quality), the M-SD cut-off should be used, whereas the M+SD cut-off should be used for matured (high-quality) courseware. In the current study, even with the high-quality courseware where only 10-20% of all instructional components in the courseware are ineffective, RAFINE was able to correctly include ineffective components in the recommendation with the M+SD cut-off.

7. DISCUSSION AND LIMITATIONS

In the evaluation study, we had two research questions. RQ1: How robust is the converse policy as a detector for relatively ineffective instructional components against different conditions of learning data? RQ2: How accurately does the frequency heuristic compose a recommendation?

First, the comparison of the normalized frequency revealed that relatively ineffective instructional components tend to appear in a policy action set significantly more frequently than effective ones regardless of the contrast and the quality of courseware. This suggests that the converse policy as a detector for relatively ineffective instructional components is robust enough against different conditions of learning data (RQ1)

Second, we evaluated the accuracy of the recommendation created by the frequency heuristic to answer RQ2. The results showed that when we use a different cut-off depending on the maturity of courseware, the recommendation created by the frequency heuristic accurately includes ineffective instructional components.

The results from the evaluation study showed that RAFINE can find deficits of the existing courseware by analyzing learning trajectory data on behalf of human experts. Although videos, quizzes and hints are evaluated in the evaluation study, RAFINE could also analyze other types of instructional components such like written paragraphs, tables, figures, etc. However, accurately tracking how students review these instructional components while learning is not straightforward—e.g., the ordinal clickstream data do not convey whether a student was reading a text instruction or not.

One limitation of this study is that there are several assumptions about learning trajectory data. First, we assume the presence of the KC model. Instructional components should be tagged with a KC to apply the RAFINE method. Therefore, the recommendation created by RAFINE changes depending on a KC model. Methods to build a good cognitive model that captures the fine KC model are studied as mentioned in the related works.

Second, we also assume that the students' mastery level is measured correctly. Since the reward function depends on the change of mastery level from the current state to the next state, it is essential that the measured mastery level is not far from the actual level of students' understanding on a skill.

Third, variations in the learning trajectory graph are critical when applying the RAFINE method. To get better performance, RAFINE must be fed a learning trajectory graph that contains diverse learning activities. If there is only one path in a learning trajectory graph, for example, the converse policy has no choice but to select an instructional component that appears in the path as a converse policy action.

One question that is not addressed in the current study is about the students' differences—how much the students' individual differences affect the “effectiveness” of each instructional component. Instructional components that are quite effective for one group of students may not be as effective for another group of students. Although it is out of the scope of the current paper, we have two working hypotheses for future studies. One hypothesis is about the majority rule—the big data overrides the individual human factors and detects the latent trends. Another hypothesis is about the individualized student model—entering individual student factors into the student model used to compute the mastery level, e.g., the individualized additive factor model [24]. Further studies will be necessary to address these issues in detail.

8. CONCLUSION

We found that the RAFINE method could serve as a building block for the evidence-based, human-in-the-loop, iterative online courseware learning engineering method by detecting the deficits of the courseware. RAFINE analyzes learning trajectory data collected from existing online courseware using the reinforcement learning technique and identifies ineffective instructional components. The detected components are provided to courseware developers as a recommendation for refinement. Given the recommendation, courseware developers can efficiently improve the courseware by modifying the listed instructional components.

In addition to providing a new evidence-based learning engineering method, we also proposed a technique called the *frequency heuristic* and contributed to the community of applications of reinforcement learning (RL). The frequency heuristic is a novel way of interpreting the policy for evaluating the actions in MDP. It operates differently from the conventional applications of RL in which the policy is used for optimization. In RAFINE, the frequency heuristic is applied to the converse policy to detect ineffective instructional components (i.e., action) that had relatively less contribution to learning. In the evaluation study, we demonstrated that the frequency heuristic over the converse policy is potentially a powerful analytic tool to detect a relative weakness among available actions.

For future studies, it is crucial to measure the actual effectiveness of the proposed method in authentic learning settings and apply the method to real students' learning data.

9. ACKNOWLEDGEMENTS

This study was partially supported by National Science Foundation grant Award No. 1623702.

10. REFERENCES

- [1] Aleven, V., & Koedinger, K. R. (2000). Limitations of student control: Do students know when they need help? In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *Proceedings of 5th International Conference on Intelligent Tutoring Systems* (pp. 292-303): Springer Verlag.
- [2] Bodily, R., Nyland, R., & Wiley, D. (2017). The RISE Framework: Using Learning Analytics to Automatically Identify Open Educational Resources for Continuous Improvement. *The International Review of Research in Open and Distributed Learning*, 18(2).
- [3] Cen, H., Koedinger, R. K., & Junker, B. W. (2006). Learning Factors Analysis - A General Method for Cognitive Model Evaluation and Improvement. In *International Conference on Intelligent Tutoring Systems* (pp. 12).
- [4] Chi, M., VanLehn, K., Litman, D., & Jordan, P. (2011). An evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach. *International Journal of Artificial Intelligence in Education*, 21(1-2), 83-113.
- [5] Clark, R., & Mayer, R. E. (2003). *e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning*. San Francisco, CA: John Wiley & Sons.
- [6] Dede, C., Richards, J., & Saxberg, B. (2018). *Learning Engineering for Online Education: Theoretical Contexts and Design-based Examples*: Routledge.
- [7] Du, X., & Cardie, C. (2017). Identifying where to focus in reading comprehension for neural question generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2067-2073.
- [8] Fishman, B., Marx, R. W., Blumenfeld, P., Krajcik, J., & Soloway, E. (2004). Creating a Framework for Research on Systemic Technology Innovations. *The Journal of the Learning Sciences*, 13(1), 43-76. doi:10.2307/1466932
- [9] González-Brenes, J. P., & Mostow, J. (2012). Dynamic Cognitive Tracing: Towards Unified Discovery of Student and Cognitive Models. *International Educational Data Mining Society*.
- [10] Guàrdia, L., Maina, M., & Sangrà, A. (2013). MOOC design principles: A pedagogical approach from the learner's perspective. *elearning papers*(33).
- [11] Iglesias, A., Martínez, P., Aler, R., & Fernández, F. (2009). Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems. *Knowledge-Based Systems*, 22(4), 266-270.
- [12] Koedinger, K. R., Corbett, A. T., & Perfetti, C. (2012). The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cognitive Science*, 36, 757-798. doi:10.1111/j.1551-6709.2012.01245.x
- [13] Lindsey, R. V., Khajah, M., & Mozer, M. C. (2014). Automatic discovery of cognitive skills to improve the

- prediction of student learning. In *Advances in neural information processing systems*, 1386-1394.
- [14] Martin, B., Mitrovic, A., Koedinger, K. R., & Mathan, S. (2011). Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-Adapted Interaction*, 21(3), 249-283. doi:10.1007/s11257-010-9084-2
 - [15] Mazidi, K., & Tarau, P. (2016). Automatic question generation: from NLU to NLG. *International Conference on Intelligent Tutoring Systems*, pp.23-33.
 - [16] Milligan, S. K., & Griffin, P. (2016). Understanding learning and learning design in MOOCs: A measurement-based interpretation. *Journal of Learning Analytics*, 3(2), 88-115.
 - [17] Rafferty, A. N., Brunskill, E., Griffiths, T. L., & Shafto, P. (2015). Faster Teaching via POMDP Planning. *Cogn Sci*, 40(6), 1290-1332. doi:10.1111/cogs.12290
 - [18] Shapiro, H. B., Lee, C. H., Wyman Roth, N. E., Li, K., Çetinkaya-Rundel, M., & Canelas, D. A. (2017). Understanding the massive open online course (MOOC) student experience: An examination of attitudes, motivations, and barriers. *Computers & Education*, 110, 35-50. doi:<http://dx.doi.org/10.1016/j.compedu.2017.03.003>
 - [19] Shen, S., & Chi, M. (2016). Reinforcement Learning: the Sooner the Better, or the Later the Better? *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. ACM, pp.37-44.
 - [20] Slavich, G., & Zimbardo, P. (2012). Transformational Teaching: Theoretical Underpinnings, Basic Principles, and Core Methods. *Educational Psychology Review*, 24(4), 569-608. doi:10.1007/s10648-012-9199-6
 - [21] Tetreault, J. R., & Litman, D. J. (2008). A reinforcement learning approach to evaluating state representations in spoken dialogue systems. *Speech Communication*, 50(8-9), 683-696.
 - [22] Wang, P., Rowe, J. P., Min, W., Mott, B. W., & Lester, J. C. (2017). *Interactive Narrative Personalization with Deep Reinforcement Learning*. Paper presented at the IJCAI.
 - [23] Wiering, M., & van Otterlo, M. (Eds.). (2012). *Reinforcement Learning*. Heidelberg, Berlin: Springer.
 - [24] Yudelson, M., Koedinger, K., & Gordon, G. (2013). Individualized Bayesian Knowledge Tracing Models. In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), *Artificial Intelligence in Education* (Vol. 7926, pp. 171-180): Springer Berlin Heidelberg.
 - [25] Zhang, Y., Shah, R., & Chi, M. (2016). Deep Learning+ Student Modeling+ Clustering: A Recipe for Effective Automatic Short Answer Grading. *International Educational Data Mining Society*.

Recommending Remedial Readings Using Student Knowledge State

Khushboo Thaker, Lei Zhang, Daqing He, Peter Brusilovsky
School of Computing and Information
University of Pittsburgh
Pittsburgh, PA, USA
k.thaker, lez39, dah44, peterb@pitt.edu

ABSTRACT

Assessment plays a vital role in learning, as it provides both instructors and students with feedback on the overall effectiveness of their teaching or learning. However, when a student fails to correctly answer certain questions in an assessment (such as a quiz), the student needs specific recommendations that are tailored to their learning needs and to the knowledge deficiency exposed by the assessment outcomes. In this paper, we explore the methods for automatically identifying the recommended textbook materials that are most relevant and suitable to the student. In particular, we conducted experiments on how to incorporate students' current knowledge state on domain concepts associated with the activity to recommend personalized remedial sections to each student. The results show that incorporating student knowledge states can significantly improve the quality of recommendations as compared to traditional content-based recommendations.

Keywords

Remedial Recommendation, student Modeling, domain concepts, dynamic student knowledge

1. INTRODUCTION

Along with the rapid development of internet and communication technologies, as well as the increasing amount of online materials in diverse formats, online learning and its supporting platforms have become vital for learning various new subjects. Regardless of if it is a self-regulated platform or instructor-regulated platform, learners are provided with diverse types of content, which may include notes, textbooks, videos, and other lecture material. Similar to traditional learning, in order to evaluate a learner's progress through course materials, various forms of assessments are embedded in the online learning process. For example, course platforms integrate quizzes and exams at the end of each learning module (section, subsection, or part). This is particularly important in self-regulated online courses, since these assessments

help learners to reflect on the content and estimate their learning progress. A complete learning loop should incorporate the provision of providing learners' relevant remedial content materials to compensate for the knowledge deficit exposed by the assessment. In classic computer-assisted instruction (CAI), where the course content and assessments were created either by the same author or by a team, links to remedial content were created manually. However, modern online learning extensively uses open educational resources and question banks created by many independent authors. In this context, an automatically generated remedial recommendation of learning content after a failed assessment is vital to the success of online learning.

A natural approach for an educational recommender system is to use content similarity as the basis for remedial recommendation [36]. This approach recommends remedial content that is similar to the assessed content. However, deficiencies in student knowledge that are exposed by the assessment might not be limited to the most similar content. Thus, a content similarity-based approach could lead to a recommendation of materials that have either already been mastered by the student, or a recommendation of material for which students lack the prerequisite knowledge.

Q: A person searches for "Michael Jordan sport" in google search engine. Please mark all the possible information needs of the person:

- A.** A person who is part of some sport
- B.** A sport named Michael Jordan
- C.** Michael from country Jordan

Figure 1: Quiz example

The goal of this study is to explore the method of remedial recommendation that dynamically address student needs. The proposed approach is to focus on modeling the domain-relevant concepts that the student is learning. The motivation for using domain-specific concepts in representing recommended documents is illustrated by Figure 1. The figure shows an example question from the "information retrieval" course. A term-based recommendation (keyword-

Khushboo Thaker, Lei Zhang, Daqing He and Peter Brusilovsky
"Recommending Remedial Readings Using Student's Knowledge state" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 233 - 244

based) could count repeated words in the question and bring up documents related to “Michael Jordan” or “Sport”. However, these documents may be irrelevant and thus would not be helpful for remediation. In contrast, a recommendation algorithm that attends to domain-specific concepts would select documents that are related to “Google search engine” or “information need”, which can cause the recommended documents to be more directly relevant to the knowledge assessed in the question.

We believe that a beneficial remedial recommendation also needs to dynamically model the student’s knowledge of the domain-specific concepts. A content-based or domain concept-based static remedial recommendation would provide the same recommendation to different students, regardless of the individual students’ different levels of knowledge deficiency exposed by the assessment outcomes. For instance, still considering the example in Figure 1, students who are already familiar with the concept of “search engine” but are struggling with the concept of “information need” would need different remedial recommendations than those who are struggling with the concept of “search engine”.

As a result, in this paper we will investigate the effects of incorporating both domain knowledge and student knowledge in remedial recommendations. More specifically, we aim to address the following research questions:

- **RQ 1:** Does the domain-based representation of educational content help perform remedial recommendation, either by acting alone or in combination with the content-based recommendation?
- **RQ 2:** Can we use automated keyphrase extraction techniques to generate domain-based representation?
- **RQ 3:** Does the augmentation of student knowledge on domain-based recommendation help in providing dynamic remedial recommendations?

To address the research questions, we proposed a concept-level static remedial recommendation (StatRemRec) and a knowledge-level dynamic remedial recommendation (DynRemRec). To conduct this research, we used an online reading platform (ReadingCircle) [13]. The system provided the students with a platform to read their course textbook. In ReadingCircle, every subsection contains a quiz to test student performance (for details 4.1). Data obtained from the ReadingCircle platform was used to investigate and evaluate the proposed recommendation approach. We hypothesize that the StatRemRec and DynRemRec approaches will provide better recommendations than state-of-the-art recommendation models that completely ignore both the domain and student dynamic knowledge states.

We also released a dataset of annotated questions in our existing online textbook with relevant sections. This data can work as a benchmark for content recommendation and linking task¹. The code and student model are available at: <https://github.com/khushsi/RemRec>

¹<https://pslcdatashop.web.cmu.edu/Project?id=637>

2. RELATED WORK

Despite their overall similarity, the roots of both static and dynamic remedial recommendation approaches can be traced to two different research areas. The *static recommendation* of educational resources does not depend on the state of an individual student, and as a result, can be generated before a student starts working with learning content. Historically, static recommendations were explored in the field of educational hypermedia and called “intelligent hypertext”, since this approach recommended resources that were not connected by a human-authored link. Research on intelligent hypertext started in the early days of this field and originally focused on linking resources using term-based resource similarity [22, 41]. Simple keyword-based approaches have been gradually replaced by semantic-level similarity, based on concepts of semantic web and domain ontology [8, 28], and later by modern text-processing approaches, such as topic modeling and concept extraction [24, 1, 37, 14].

The emergence of MOOCs and the online accumulation of large volumes of educational content encouraged a new wave of research on “intelligent” linking focused on connecting primary learning content, such as textbooks and MOOCs, with different external learning resources, such as videos, Wikipedia pages, or research papers [1, 18, 20].

In contrast, the *dynamic recommendation* model of educational content has to be generated on the fly, based on the current knowledge or interest of the learner. Dynamic recommendations could be traced back to the classic works on adaptive course “sequencing” [23] and generation [10]. The first generation of this work focused on adaptation to student levels of knowledge and used different student modeling approaches from the field of intelligent tutoring [35]. The emergence of recommender systems encouraged a different generation of research on dynamic recommendation that focused on learner interests and used techniques from the areas of recommender systems [21]. Due to its popularity, the term “recommendation” is now used to refer to both knowledge-based and interest-based recommendations. Recent work on educational recommendation frequently combines both knowledge and interest adaptation and supports a range of needs, such as fine-grained resource recommendation for practice activities [2, 37], reading materials [29], and videos, as well as coarse-grained recommendation of courses [30, 7] or textbooks [31].

The majority of research on educational recommendation has focused on recommending students’ next thing to do and assumes that the student’s overall progress is good. A different recommendation approach, known as remedial recommendation [3], has focused on recommending resources that can help a student to learn a concept in which a student is weak, in order to improve understanding or resolve misconceptions. Konstantin et al. [4] proposed a knowledge-gap based remedial recommendation approach. The method considers learners’ previous success rates and categorize learners as expert, intermediate, or unknown. They found that this coarse-grained categorization may help in providing recommendations based on student needs. Although such a coarse-grained categorization is beneficial, it assumes that there is a single learning rate for all students. However, the existing advancement in education technologies have ways

to infer students' individualized levels of knowledge [32] and learning rates [9, 11], called student models. In our work, we used student models to define fine-grained student knowledge states and explored the possibility of using them in remedial recommendations.

3. METHODOLOGY

In this paper, we investigated the effect of dynamically incorporating domain knowledge and student knowledge for remedial recommendation. The intuition is that this dynamic incorporation can enable more relevant and suitable resources to help students recover from failure within the assessment.

3.1 Problem Description

Formally, the research problem can be described as: for a given student S , who was recently assessed on question q ; if student S failed on question q , we want to provide student S with a recommended reading from a content set $T = t_1, t_2, \dots, t_c$ to help the student to grasp the knowledge that would be required to succeed on question q .

The vector representation of texts T and question q are constructed using domain concepts, the recommendation is computed based on the cosine similarity between the vectors of text T and question q , and the top five most similar texts are selected for the recommendation.

3.2 Static Remedial Recommendation (StatRemRec)

StatRemRec targets remedial recommendation based on incorporating semantic knowledge or domain knowledge of the content. To build a domain-based representation of education material, we used domain concepts. The approach of a domain concept-based representation of education material is commonly found in intelligent tutoring systems, which focus on problem-solving support and where every practice problem is associated with a set of domain knowledge components (concepts) [17]. In our case, concepts are expressed as key phrases. Each key phrase depicts a fragment of domain knowledge, a semantic entity, or a fine-grained topic. Each target education material, textbook section, and question (in our case) are annotated with domain concepts. Figure 2 shows an example output of these annotated domain concepts mapped to both a text and a question.

Once we have obtained a domain concept for both texts and questions, we build a representation of texts and questions as a frequency-based vector representation, based on the presence of domain concepts for each text and each question. For recommending texts for a particular question, we apply cosine similarity between question q and all the available sections in the text T and rank the top five most similar texts $\{R_q^1, R_q^2, R_q^3, R_q^4, R_q^5\}$ that share the same domain concepts with the questions q .

3.3 Dynamic Remedial Recommendation (DynRemRec)

StatRemRec accounts for semantic knowledge in the document for recommending remedial materials. Although this is an improvement on a purely keyword-based content recommendation system, StatRemRec still recommends the same

content for each student, regardless of the student's real-time content requirement. For example, a student failing on a question that asks about "Multiplication" will always be given recommendations for readings related to "Multiplication" with StatRemRec. For instance, if a student's skills are weak in a prerequisite concept, e.g. "Addition", it is crucial to support that student's current needs.

In education systems, intelligent tutoring systems account for this student-specific information to provide students with adaptive practices and has been shown to help with effective and efficient learning [38]. To provide adaptation, the tutors maintain dynamically changing student knowledge states while the student uses the tutoring system.

In DynRemRec, we maintained dynamically changing student knowledge states and used students' real-time knowledge states to generate a personalized remedial recommendation for each student. The following subsections provide details about student knowledge state generation and our approaches to integrating them into our remedial recommendation.

3.3.1 Student Knowledge State Generation

For generating students' knowledge state, we used a traditional and widely accepted student modeling framework, performance factor analysis (PFA) [34]. This model relies on expert annotated *skills* (also known as knowledge components or concepts). *Skills* are knowledge units associated with student activities, steps, and questions on which students' knowledge and performance are tested [17]. In our work, we considered *domain concepts* as *skills*, which has been shown to work in previous work on student modeling in online textbooks. [40, 16, 42]. At the base of the model is a Qmatrix, a binary matrix where columns represent *concepts or skills* and rows represent questions. Each cell is a binary value, where 1 in the cell with row r and column c represents that question r is an application of concept c . PFA represents the student's probability of success in answering a question as a function of the student's previous successful and failed attempts on the *concept* associated with the question, as shown in Equation 1

$$\text{PFA: } \ln \frac{p_{sq}}{1 - p_{sq}} = \alpha_s + \sum_c \beta_c Q_{cq} + \sum_c Q_{cq} (\mu_c S_{sc} + \rho_c F_{sc}) \quad (1)$$

where, s is a student and q is a question. c is a *concept* (skill or knowledge component). α_s is a coefficient associated with learner s (regression intercept) and represents the proficiency of learner s . Q is a Qmatrix and Q_{cq} is the Qmatrix cell associated with question q and *Concept* c . β_c are coefficients associated with concept c . β_c represents the difficulty of concept c , while μ_c and ρ_c are coefficients associated with S_{sc} and F_{sc} . S_{sc} and F_{sc} are the number of success and failure attempts, respectively, of learner s on concept c . We consider PFA, as PFA provides granular evaluation based on individual students' prior success and failure on a particular skill [34].

Example Text:

[Information retrieval](#) is the activity of obtaining [information system](#) resources that are relevant to an [information need](#) from a collection of those resources. [Searches](#) can be based on full-text or other [content-based indexing](#). Information retrieval is the science of searching for information in a document, searching for documents themselves, and also searching for the [metadata](#) that describes data, and for databases of texts, images or sounds.

Example Question:

Which technique discards information while applying [compression](#)?

A) [Lossless compression](#). B) [Tiny compression](#). C) [Zip compression](#). D) [Lossy compression](#)

Figure 2: An example of domain concepts annotated to both a text and a question. This is just for example purposes and is not part of our study. The domain concepts are marked with a link

To obtain the knowledge of a student s on a concept c when the student failed on question q , we generate the probability of failure on a concept PF_{sc} . We assumed that there exists an item c annotated with concept c , and generated the probability of failure as:

$$PF_{sc} = \begin{cases} 1 - (\alpha_s + \beta_c + (\mu_c S_{sc} + \rho_c F_{sc})) & c \in q \\ 0 & c \notin q \end{cases} \quad (2)$$

where α_s is ability of student s and β_c is difficulty of concept c . S_{sc} and F_{sc} are previous success and failed attempts of student s on concept c , after the student failed on question q .

To generate the student knowledge state vector, we represented each domain concept associated with the question q with weight value PF_{sc} . The knowledge state consists of the probability of failure to make sure that a greater weight is given to domain concepts (*skills*) where the student has a high probability of failure (where they might lack sufficient knowledge). For the concepts that are not associated with question q , we made the probability to be zero, as the goal is to recommend material related to concepts that are associated with the question.

The representation of text (documents or textbook sections) is the same for both DynRemRec and StatRemRec (i.e. representation based on frequency on domain concepts, as discussed in Section 3.3). The change is in question representation, which is based on the presence of domain concepts in StatRemRec and is based on the dynamic knowledge state of domain concepts in DynRemRec.

4. EXPERIMENTS

The dataset from ReadingCircle [13] was used for exploring DynRemRec and StatRemRec. In this section, we will introduce the dataset before presenting the details of our experiments.

4.1 Student Dataset

ReadingCircle [13] is an online reading platform. It provides an online reading environment to students in a course where they read assigned textbook materials to prepare for class.

There are quizzes of questions embedded in each section of the assigned readings to assess the progress of student learning on the content.

ReadingCircle keeps extensive logs for events associated with student reading and assessment. The dataset used in the experiments is collected from a version of ReadingCircle that has been adapted for supporting a graduate-level course on information retrieval at an University of Pittsburgh in spring 2016. There was no restriction on the number of attempts to the questions. ReadingCircle logs each and every attempt made by the student. This data set contains 9006 quiz interactions from 22 students and 4273 interactions of student failure on quizzes (for more details, refer to Table 1). The student dataset can be obtained from Datashop².

Table 1: ReadingCircle data details

Number of documents (sections)	66
Number of questions	89
Number of students	22
Average per student questions attempted	91
Student practice interactions	9006
Number of failure interactions	4273

4.1.1 Ground Truth

To evaluate the effect of recommendations on questions, we require a mapping of questions to sections. Each question maps to the section where it appears. This assumption holds as each quiz is created by subject experts (instructors and teaching assistants) to assess student knowledge in a particular section. In a few cases, a quiz will assess multiple sections. In this case, we map the questions appearing in those quiz to multiple sections. For more details on the question-to-section mapping dataset, please refer to Table 2

Table 2: Ground truth for recommendation

Number of (sections)	66
Number of questions	89
Number of questions per section	1.93
Number of questions linked to single section	81
Number of questions linked to multiple sections	8

²<https://pslcdatashop.web.cmu.edu/Project?id=637>

4.2 Domain Concept

Concept-based textbook representation was introduced by early projects that focused on adaptive textbooks [15, 43]. Adaptive textbooks associated every section of a digital textbook with a set of domain concepts (called outcomes) that are present in that section. In this work, we investigated both expert-annotated *domain concepts* [42] and automated extracted *domain concepts* [26, 6, 25].

- **Expert-based concepts (EBC):** For EBC, we used concepts that were generated by Wang et al [42]. Wang et al. [42] developed comprehensive expert-based annotation rules and proposed a two-step concept annotation system with three subject experts. [42]. The concepts are available for sections in the “Introduction of Information Retrieval” book, which is the same book that students are reading in the online course in ReadingCircle.

In order to conduct the experiment, we want both questions and the text that are associated with the concepts. However, EBC is only available for textbook sections. In order to associate concepts with questions, we created a list of domain concepts using concepts in all sections, and performed a simple lookup on the concept list to find the domain concepts in the question and answer text. More details about the EBC concepts is mentioned in Table 3.

Table 3: EBC Concepts

Number of unique concepts	1047
Average number of concepts per section	30.83
Average number of concepts per quiz	6.52

To check if questions have concepts only from related sections, we plotted the distribution that depicts the number of unique sections that share concepts with questions, as shown in Figure 3. These statistics show that questions share concepts with an average of 24.3 sections.

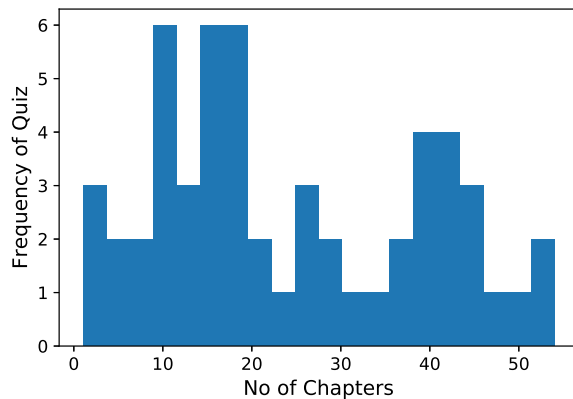


Figure 3: Distribution of number of unique sections sharing domain concepts per question.

- **Automated concept extraction (ACE):**

StatRemRec and DynRemRec represent text with concept-level representation. In the case of DynRemRec, the student model is also trained using concepts associated with student practice activities. However, traditional expert-based concept generation is time consuming and hard to obtain with the incorporation of open-source course materials. Hence, it is impractical if time-consuming expert concepts becomes a necessary step in student recommendation on education resources. To prove the ease of incorporating StatRemRec and DynRemRec, we dedicated this set of experiments to test the feasibility of concept-level remedial recommendations (RQ 2). The experiment was designed to test StatRemRec and DynRemRec on concepts that were automatically extracted through keyphrase extraction techniques [6, 25, 26]. It is debatable if keyphrase extraction techniques extract domain concepts and could be used as knowledge components or *skills* on which students’ knowledge is measured. For evidence of using concepts as *skills*, we relied on evidence from work by Thaker et al. [40] and Huang et al. [16], in which student models are trained on keyphrases in education content for adaptive textbooks.

To perform the experiment and explore research question RQ-2, we selected three state-of-the-art keyphrase extraction techniques, as discussed below:

1. TextRank: TextRank [26] is a classic unsupervised keyphrase extraction technique. TextRank converts each document into a graph of words, based on word co-occurrence criteria. The algorithm then applies the page rank algorithm to the graph and extracts the important keyphrases.
2. CopyRNN: CopyRNN [25] is a supervised deep-learning based sequence-to-sequence keyphrase generation technique. CopyRNN is one of the state-of-the-art supervised keyphrase extraction techniques. This will help us evaluate our model for supervised keyphrase extraction.
3. TopicRank: TopicRank [6] is a graph-based unsupervised keyphrase extraction technique. The difference is TopicRank focuses on finding keyphrases that belong to the topic of the document. As a result, this technique can provide more insight into topic-based concept extraction.

Table 4 shows more details of the concepts extracted by different ACE methods. The table indicates that different algorithms will choose a different domain space for representing the domain.

4.3 Term-Based Recommendation Baseline (TextRec)

For our baseline, we used a simple term-based recommendation approach. TextRec finds the similarity between two documents (section text and quiz text) based on words that are present in the text. Each term in the document is used as a semantic unit and the document is represented as a vector of the TF-IDF weights [39]. Such a TF-IDF based document similarity was recently found to be effective for

Table 4: Statistics of ACE datasets

Model	Average no of Concepts Section	Quiz	Unique Concepts	Overlap Sections
CopyRNN	14.56	2.32	558	2.29
TextRank	27.39	7.53	698	28.88
TopicRank	96.01	7.85	2469	32.94

finding similar education resources [37]. Although state-of-the-art recommendation techniques use advanced semantic representations that use both word and document embeddings [27], we did not explore much in this area, as our focus is in understanding the effectiveness of including domain and student knowledge in recommending remedial resources.

4.4 Experiment Steps

To address the research questions mentioned in Section 1, we conducted the following experiments:

1. **Term vs Concept Representation:** To understand if domain-based representation is effective (RQ 1), we compared the concept-level techniques of StatRemRec and DynRemRec to the term-based approach TextRec.
2. **Fusion experiment:** As the TextRec approach has a term-based representation of education material, while StatRemRec and DynRemRec use concept-level representation, to leverage both types of representation (RQ 1), we fused the term-based approach with domain concept-based approaches. The fusion of term-based methods with concept-based methods is done by simple linear interpolation, as specified in Equation 3

$$Sim_{q_i, t_j}^{fused} = \alpha \cdot Sim_{q_i, t_j}^{concept} + (1 - \alpha) \cdot Sim_{q_i, t_j}^{text} \quad (3)$$

where Sim_{q_i, t_j} is the similarity between question q_i and section t_j . To determine the interpolation coefficient α in Equation 3, we selected the α that gave the best result for *TextRec + StatRemRec* on expert-based concepts and used it in all of our experiments.

3. **Experiment with ACE:** To address research question RQ 2, we performed remedial recommendation by using keyphrases as domain concepts.
4. **Knowledge Augmentation:** To address research question RQ 3, this experiment investigated differences in the recommendations generated from both StatRemRec and DynRemRec.

Figure 4 provides a complete picture of the experiment set up with all of the resources that were used for the experiment. Student interaction data is divided into students stratified in ten random folds. The training folds are used for training the student model, with available concept indexing for each question. The recommendation is evaluated on the test fold. The student model is used to generate dynamic knowledge-based concept representations for DynRemRec, and the results reported in Section 5 are averaged over 10 test folds.

4.5 Evaluation Metric

As discussed in Section 4.1.1, the question to section mapping is one to many, so we adopted mean reciprocal rank (MRR) and mean average precision (MAP) to evaluate the recommendations [33]. MRR is a good metric to understand, on average, the position on which a relevant recommendation is obtained, and MAP@5 will generally prefer the algorithm that recommends more relevant sections at the top of the list. Here, the wrong recommended section is considered not to be relevant and the correct section is considered to be relevant.

5. RESULTS AND DISCUSSION

5.1 Term vs Concept-Based Recommendation

Table 5 shows the performance of the term-based approach TextRec the and domain concept-based approaches StatRemRec and DynRemRec. Both StatRemRec and DynRemRec performed lower than the baseline TextRec. One potential reason for this finding is that TextRec relies on the keywords from the whole content of both the quiz and the section, while both StatRemRec and DynRemRec only index based on a small number of identified concepts. Based on our calculation, the average length of sections in our dataset is 1,345 words, whereas the average number of concepts annotated by experts in each section is only 13.5, which indicates that the concept-based representation relies on a comparatively few number of concepts.

These results show that, despite the importance of domain-specific concepts in explaining the content in education, confining the representation of text with only concept-level content could cause too much loss in useful textbook content.

5.2 Fusing Term and Concept-Based Recommendations

As Meng et al. [24] pointed out, term-based content representations of education materials can provide fine-grained term level information and statistics, while concept-based representation works on both a coarse-grained topic and semantic level. Consequently, it is beneficial to have information from both these representations when recommending a relevant document to a student. As term-based representation identifies the content similarity based on shared terms, concept-level representation provides emphasis on semantics and the knowledge that is represented by the concepts. To leverage the combined power of these two representations, we conducted fusion experiments on both the term- and concept-level approaches, as mentioned in Section 4.4. As Table 5 shows, there is a clear indication that fusion (*TextRec + StatRemRec*) surpasses the baseline TextRec and benefits from concept-level representation. To investigate this effect in detail, we plotted the performance of StatRem-

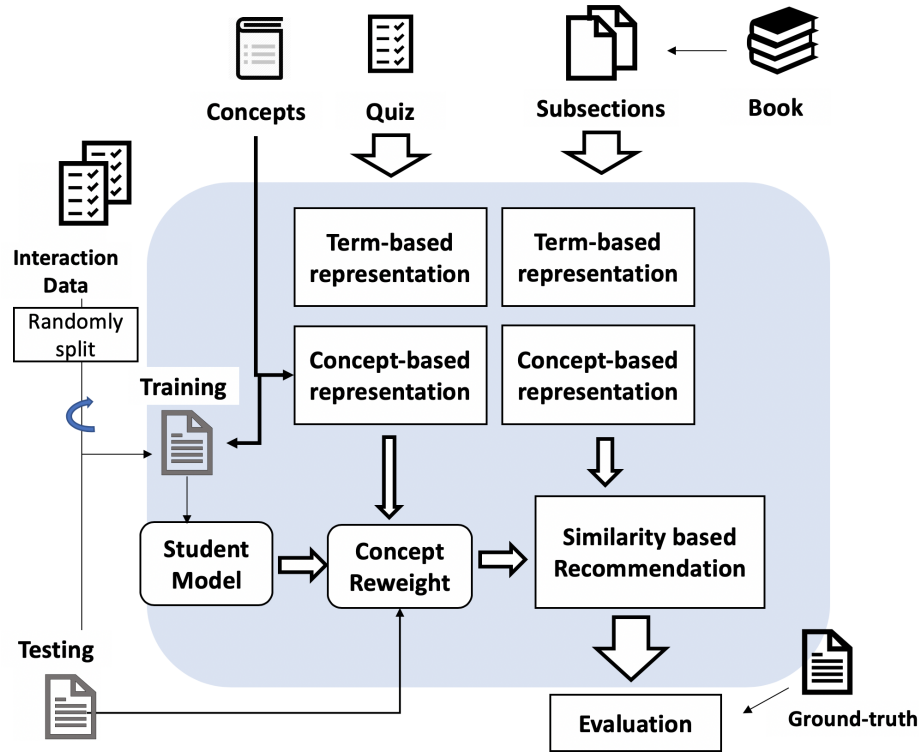


Figure 4: Experiment Setup

Table 5: Remedial recommendation performance on failed questions in ReadingCircle system in terms of MRR. * denotes a significant performance change of *metric* over text-based recommendation TextRec using a non-parametric Wilcoxon signed rank test. Numbers shown in bold indicate the top two best performance. The performance is based on expert annotated EBC. Parameter α for *TextRec + StatRemRec* and *TextRec + DynRemRec* is 0.60 based on *MAP@5*

Model	student knowledge	MRR	MAP@5
TextRec	-	83.00	74.01
StatRemRec	-	73.47	68.40
DynRemRec	✓	71.05	66.06
TextRec + StatRemRec	-	*91.01	*86.18
TextRec + DynRemRec	✓	*89.53	*83.90

Rec for different values of the interpolation co-efficient α , as shown in Figure 6.

Figure 6 and Figure 5 display some important characteristics of these recommendations. The curve of *TextRec + StatRemRec* starts with the performance of TextRec at the value of $\alpha = 0$. Initially, along with α increases from $\alpha = 0$ to $\alpha = 0.3$, the performance of the fusion-based approach *TextRec + StatRemRec* increases both in MRR and MAP@5, which shows the benefit from the inclusion of concept-based representation. Next, MRR performance stabilizes for a period from $\alpha = 0.3$ to $\alpha = 0.6$, which shows no obvious change in the position of the top-ranked relevant documents in rank lists. In this interval, MAP@5 (Figure 6) keeps increasing, which shows that concept-level representation are helping in either recommending new relevant documents or bringing already ranked relevant documents up at a higher

ranked position. The performance of MAP, which looks at all the recommendations, is best at $\alpha = 0.60$. Since the performance improvement comes with the increasing weight on StatRemRec, it provides evidence that recommendation benefits more from the domain-specific concept-based representation. Fusion improved the performance of recommendation by 16% (significantly, with significance tested using Wilcoxon signed rank test), providing an answer to our research question RQ 1 that StatRemRec improves recommendation quality when augmented with a traditional content-based recommendation system.

5.3 Performance with ACE

In research question RQ 2, the goal is to understand the feasibility of using ACE as a domain concept and to use it in providing remedial recommendations. Table 6 compares the performance of some traditional ACE techniques.

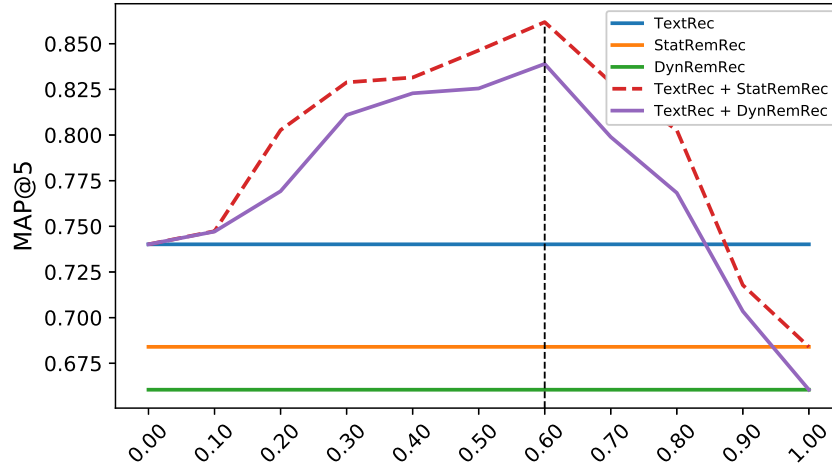


Figure 5: Threshold based recommendation values in terms of MAP with change in α values

We experimented with both supervised and unsupervised keyphrase extraction techniques. It is apparent from the performance of ACE that these techniques beat traditional content-based recommendation systems. TextRank does not improve much on content-based recommendation, but both TopicRank and CopyRNN surpass content-based recommendation. TopicRank is a winner among three methods and beats the recommendations that are based on EBC. A possible reason could be the difference between keyphrase extraction methods. TextRank and CopyRNN focus on extracting important keyphrases from a document, while TopicRank is focused on extracting keyphrases that are related to topics discussed in a document. Thus keyphrase extracted by TopicRank is more analogous to concepts discussed in the course. We experimented with comparatively few and somewhat simple keyphrase extraction techniques, as we aim to provide a piece of simple evidence for the feasibility of our approach (RQ 2). We leave for future work to perform a more comprehensive experiment with automated concept extraction, which extracts more advance domain knowledge like prerequisites and outcomes within a textbook [12, 19].

5.4 Augmenting Knowledge in Concept-Level Representation

The main goal of DynRemRec is to provide students with personalized remedial recommendations based on their real-time information needs. As Table 5 shows, DynRemRec performed worse than both TextRec and concept-based StatRemRec in MRR and MAP@5. As with StatRemRec, we fused DynRemRec with TextRec. The fusion of DynRemRec with term-based representation (TextRec + DynRemRec) revealed a similar output as TextRec + StatRemRec. This fusion improved the performance of recommendations by 13%, as compared to TextRec.

Although the fusion (TextRec + DynRemRec) improved the results in the case of DynRemRec, it is evident from Figure 6 that *TextRec + DynRemRec* was not able to improve over the performance of *TextRec + StatRemRec*. An explanation

of this output is that DynRemRec addresses the need of students at each recommendation, while StatRemRec provides the same static recommendation to each student. This means that there may be cases in which experts think that a student will benefit from reading a particular section, but actual student needs might differ. Our current gold standard is expert-based, which does not target real-time student needs.

5.4.1 Effectiveness of augmenting knowledge

The goal of DynRemRec is to tailor the recommendations to student needs. Figure 7 shows the distribution of a unique set of recommendations generated against each question by *TextRec + DynRemRec*. As presented in the distribution in Figure 7, except for 12 questions, all the questions generated more than one distinct ranked list of recommendations. The results indicate that knowledge augmentation helps in providing an adaptive recommendation.

To understand the difference between StatRemRec and DynRemRec, we further investigated the cases where the recommendation of StatRemRec was different from DynRemRec. In our online textbooks, students read one to two chapters every week. The course instructors predefined the course sequence. We divided the course sections into three categories: *previous sections*, *current sections*, and *future sections*, based on the section of the question for which a student received the remedial recommendation. Previous sections can be considered as prerequisite sections, while the current section is the one for which the student is assessed. Future sections are advanced topics in which students lack complete knowledge. Figure 8 shows the distribution of recommended sections based on the three remedial recommendation techniques of TextRec, TextRec + StatRemRec, and TextRec + DynRemRec. A good remedial recommendation algorithm will recommend resources from current sections and previous sections, as understanding concepts explained in both previous and current sections will help students to solve the failed question. As Figure 8 shows, TextRec’s recommendation is distributed in all the categories, while both TextRec + StatRemRec and TextRec + DynRemRec have

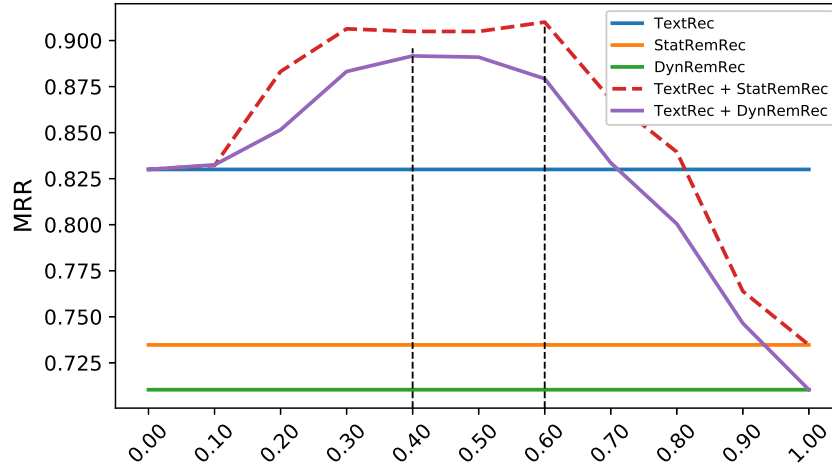


Figure 6: Threshold based recommendation metric in terms of MRR with change in α values

Table 6: Remedial recommendation performance on failed questions in an ReadingCircle system, in terms of MAP@5. The performance compares EBC with ACE-based methods for both StatRemRec and DynRemRec with an alpha value of 0.60, determined by experiment, in Figure 6. The performance is mean on 10 folds. Bold indicates the performance, which is as good as EBC.

Model	TextRec + StatRemRec	TextRec + DynRemRec
	MAP@5	MAP@5
<i>ACE</i>		
TextRank	83.14	83.97
CopyRNN	84.66	84.05
TopicRank	89.90	88.85
<i>EBC</i>	86.81	84.11

more recommendations in current sections and fewer recommendations in future sections. This result shows the clear benefit of the addition of domain knowledge, which helps in recommending the sections that are appropriate for the learner.

Augmenting student knowledge (TextRec + DynRemRec), on the other hand, further decreased the recommendation of future sections. However, DynRemRec also decreased the current section and provided more recommendations in previous sections than StatRemRec. Recommendations on previous section can be the consequence of students' knowledge state. If a student is still weak in a prerequisite concept, StatRemRec will not consider those cases, while DynRemRec, which provides adaptive remedial recommendation, will make recommendations that are based on students' needs. This gives indirect evidence about the effectiveness of augmenting student knowledge in recommending resources.

6. CONCLUSIONS AND FUTURE WORK

This paper investigated the value of using domain and student knowledge for the remedial recommendation of reading resources.

We found that the use of domain knowledge significantly improves recommendation performance when fused with traditional content-based recommendations. The model *TextRec*

+ *StatRemRec*, which augments content-based recommendation with domain concept-based recommendations, significantly outperformed the traditional content-based recommender TextRec. Currently, fusion is achieved with a simple linear interpolation; we would like to investigate other fusion techniques in future studies.

While domain knowledge improves the quality of recommendation, it doesn't account for the knowledge and needs of individual students when recommending remedial reading. To address this, we tried to use dynamic student models that represent students' current knowledge state on domain concepts for providing truly personalized recommendations. *TextRec* + *DynRemRec*, which augments student knowledge with a content-based recommender, provided evidence to support the benefits of adding students' knowledge state for an adaptive recommendation. Although we provided some preliminary evidence for a personalized recommendation, it would be necessary to conduct a comprehensive study with real-time student feedback on recommendation. In future work, we will further investigate this phenomenon by incorporating different recommendation techniques to our online course platform. Such a study will provide a more accurate evaluation based on students' learning gain and overall system usage.

To address research questions RQ 1 and RQ 3, we used

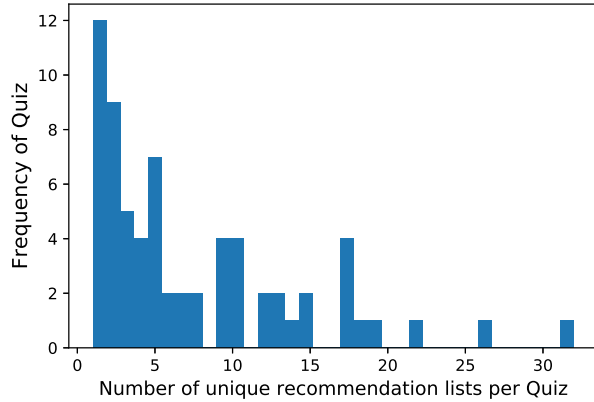


Figure 7: Distribution of unique lists of recommendations per quiz by *TextRec* + *DynRemRec*

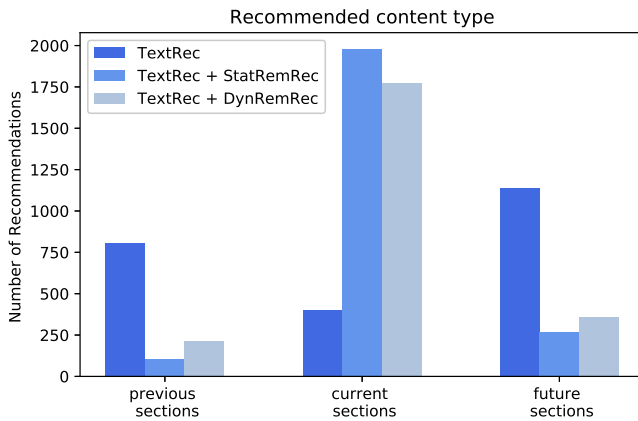


Figure 8: Category of recommended sections. The graph only plots for top recommended section. Previous, current, and future sections are categorized according to the sequence of the course.

expert-annotated domain knowledge (EBC) for building our recommender. As expert-provided concept indexing is expensive in terms of both time and resources, we further investigated traditional concept extraction approaches, such as ACE, to make our approach more feasible in practice. The performance of domain and knowledge augmented recommender on ACE proves that the technique is easy to adapt to new course content, for which expert-based concept indexing may not be available. A good future direction for this work is to investigate the importance of ACE by incorporating advanced semantic topic modeling [5] and prerequisite extraction techniques [19, 12]. A better representation of domain knowledge can lead to a more reliable knowledge unit generation for pedagogical design.

This work represents a first exploration of the power of considering students’ knowledge state in recommending personalized remedial readings. The present work provides an interesting insight into automated remedial recommendation. We believe these types of models could play a more promi-

nent role in future models of online learning where immediate or individualized instructor feedback is not available.

7. ACKNOWLEDGEMENTS

This paper was supported by the National Science Foundation Grants IIS-1525186 and Provost’s Personalized Education Grants³ by University of Pittsburgh. All data presented here is available from DataShop⁴.

8. REFERENCES

- [1] R. Agrawal, S. Gollapudi, A. Kannan, and K. Kenthapadi. Study navigator: An algorithmically generated aid for learning from electronic textbooks. *Journal of Educational Data Mining*, 6(1):53–75, 2014.
- [2] F. Ai, Y. Chen, Y. Guo, Y. Zhao, Z. Wang, G. Fu, and G. Wang. Concept-aware deep knowledge tracing and exercise recommendation in an online learning system. In M. C. Desmarais, C. F. Lynch, A. Merceron, and R. Nkambou, editors, *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada*. International Educational Data Mining Society (IEDMS), 2019.
- [3] K. S. R. Anjaneyulu, R. A. Singer, and R. Harding. Usability studies of a remedial multimedia system. *Journal of Education Multimedia Hypermedia*, 7(2–3):207–236, June 1998.
- [4] K. Bauman and A. Tuzhilin. Recommending remedial learning materials to students by filling their knowledge gaps. *MIS Quarterly*, 42(1), 2018.
- [5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- [6] A. Bougouin, F. Boudin, and B. Daille. Topicrank: Graph-based topic ranking for keyphrase extraction. In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan*, pages 543–551. Asian Federation of Natural Language Processing / ACL, 2013.
- [7] H. Bydzovská. Course enrollment recommender system. In T. Barnes, M. Chi, and M. Feng, editors, *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016, Raleigh, North Carolina, USA*, pages 312–317. International Educational Data Mining Society (IEDMS), 2016.
- [8] L. Carr, W. Hall, S. Bechhofer, and C. A. Goble. Conceptual linking: ontology-based open hypermedia. In V. Y. Shen, N. Saito, M. R. Lyu, and M. E. Zurko, editors, *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China*, pages 334–342. ACM, 2001.
- [9] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modelling the acquisition of procedural knowledge. *User Modeling User-adapted Interaction*, 4(4):253–278, 1995.
- [10] T. Diessel, A. Lehmann, and J. Vassileva. Individualized course generation: A marriage between cal and ical. In M. R. Kibby and J. R. Hartley,

³<https://www.personalized.pitt.edu/content/iris-intelligent-recommender-instructors-and-students-completing-personalized-assessment>

⁴<https://pslcdatashop.web.cmu.edu>

- editors, *Computer Assisted Learning: Selected Contributions from the CAL '93 Symposium*, pages 57–64. Pergamon, Amsterdam, 1994.
- [11] M. Dudík, S. J. Phillips, and R. E. Schapire. Performance guarantees for regularized maximum entropy density estimation. In J. Shawe-Taylor and Y. Singer, editors, *Learning Theory, 17th Annual Conference on Learning Theory, COLT 2004, Banff, Canada, Proceedings*, volume 3120 of *Lecture Notes in Computer Science*, pages 472–486. Springer, 2004.
 - [12] F. Gasparetti, C. D. Medio, C. Limongelli, F. Sciarone, and M. Temperini. Prerequisites between learning objects: Automatic extraction based on a machine learning approach. *Telematics Informatics*, 35(3):595–610, 2018.
 - [13] J. Guerra, D. Parra, and P. Brusilovsky. Encouraging online student reading with social visualization. In E. Walker and C. Looi, editors, *Proceedings of the Workshops at the 16th International Conference on Artificial Intelligence in Education AIED 2013, Memphis, USA*, volume 1009 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.
 - [14] J. Guerra, S. A. Sosnovsky, and P. Brusilovsky. When one textbook is not enough: Linking multiple textbooks using probabilistic topic models. In D. H. Leo, T. Ley, R. Klammar, and A. Harrer, editors, *Scaling up Learning for Sustained Impact - 8th European Conference, on Technology Enhanced Learning, EC-TEL 2013, Paphos, Cyprus.*, volume 8095 of *Lecture Notes in Computer Science*, pages 125–138. Springer, 2013.
 - [15] N. Henze, K. Naceur, W. Nejdl, and M. Wolpers. Adaptive hyperbooks for constructivist teaching. *Künstliche Intelligenz*, 13(4):26–31, 1999.
 - [16] Y. Huang, J. P. González-Brenes, R. Kumar, and P. Brusilovsky. A framework for multifaceted evaluation of student models. In O. C. Santos, J. Boticario, C. Romero, M. Pechenizkiy, A. Merceron, P. Mitros, J. M. Luna, M. C. Mihaescu, P. Moreno, A. Hershkowitz, S. Ventura, and M. C. Desmarais, editors, *Proceedings of the 8th International Conference on Educational Data Mining, EDM 2015, Madrid, Spain*, pages 203–210. International Educational Data Mining Society (IEDMS), 2015.
 - [17] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science*, 36(5):757–798, 2012.
 - [18] M. Kokkodis, A. Kannan, and K. Kenthapadi. Assigning educational videos at appropriate locations in textbooks. In J. C. Stamper, Z. A. Pardos, M. Mavrikis, and B. M. McLaren, editors, *Proceedings of the 7th International Conference on Educational Data Mining, EDM 2014, London, UK*, pages 201–204. International Educational Data Mining Society (IEDMS), 2014.
 - [19] I. Labutov, Y. Huang, P. Brusilovsky, and D. He. Semi-supervised techniques for mining learning outcomes and prerequisites. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada*, pages 907–915. ACM, 2017.
 - [20] X. Liu, Z. Jiang, and L. Gao. Scientific information understanding via open educational resources (OER). In R. Baeza-Yates, M. Lalmas, A. Moffat, and B. A. Ribeiro-Neto, editors, *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile*, pages 645–654. ACM, 2015.
 - [21] N. Manouselis, H. Drachsler, K. Verbert, and E. Duval. *Recommender Systems for Learning*. Springer Briefs in Electrical and Computer Engineering. Springer, 2013.
 - [22] J. T. Mayes, M. R. Kibby, and H. Watson. Strathtutor: The development and evaluation of a learning-by-browsing on the macintosh. *Computers and Education*, 12(1):221–229, 1988.
 - [23] D. McArthur, C. Stasz, J. Hotta, O. Peter, and C. Burdorf. Skill-oriented task sequencing in an intelligent tutor for basic algebra. *Instructional Science*, 17(4):281–307, 1988.
 - [24] R. Meng, Y. Huang, D. He, and P. Brusilovsky. Knowledge-based content linking for online textbooks. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016, Omaha, NE, USA*, pages 18–25. IEEE Computer Society, 2016.
 - [25] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, and Y. Chi. Deep keyphrase generation. In R. Barzilay and M. Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, Volume 1: Long Papers*, pages 582–592. Association for Computational Linguistics, 2017.
 - [26] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, Barcelona, Spain*, pages 404–411. ACL, 2004.
 - [27] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA*, 2013.
 - [28] D. N. Milne and I. H. Witten. Learning to link with wikipedia. In J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K. Choi, and A. Chowdhury, editors, *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA*, pages 509–518. ACM, 2008.
 - [29] M. Mohseni, M. L. Maher, K. Grace, N. Najjar, F. Abbas, and O. Eltayeb. Pique: Recommending a personalized sequence of research papers to engage student curiosity. In S. Isotani, E. Millán, A. Ogan, P. M. Hastings, B. M. McLaren, and R. Luckin, editors, *Artificial Intelligence in Education - 20th International Conference, AIED 2019, Chicago, IL, USA, Proceedings, Part II*, volume 11626 of *Lecture Notes in Computer Science*, pages 201–205. Springer, 2019.
 - [30] R. Morsomme and S. V. Alferez. Content-based course recommender system for liberal arts education. In

- M. C. Desmarais, C. F. Lynch, A. Merceron, and R. Nkambou, editors, *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada, July 2-5, 2019*. International Educational Data Mining Society (IEDMS), 2019.
- [31] R. Nagata, K. Takeda, K. Suda, J. Kakegawa, and K. Morihiro. Edu-mining for book recommendation for pupils. In T. Barnes, M. C. Desmarais, C. Romero, and S. Ventura, editors, *Educational Data Mining - EDM 2009, Cordoba, Spain. Proceedings of the 2nd International Conference on Educational Data Mining*, pages 91–100, 2009.
- [32] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In P. D. Bra, A. Kobsa, and D. N. Chin, editors, *User Modeling, Adaptation, and Personalization, 18th International Conference, UMAP 2010, Big Island, HI, USA*, volume 6075 of *Lecture Notes in Computer Science*, pages 255–266. Springer, 2010.
- [33] D. Parra and S. Sahebi. Recommender systems: Sources of knowledge and evaluation metrics. In *Advanced Techniques in Web Intelligence-2: Web User Browsing Behaviour and Preference Analysis*, pages 149–175, Berlin, Heidelberg, 2013. Springer.
- [34] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis - A new alternative to knowledge tracing. In V. Dimitrova, R. Mizoguchi, B. du Boulay, and A. C. Graesser, editors, *Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling, Proceedings of the 14th International Conference on Artificial Intelligence in Education, AIED, Brighton, UK*, volume 200 of *Frontiers in Artificial Intelligence and Applications*, pages 531–538. IOS Press, 2009.
- [35] M. C. Polson and J. J. Richardson. *Foundations of intelligent tutoring systems*. Lawrence Erlbaum, 1988.
- [36] B. Pursel, C. Liang, S. Wang, Z. Wu, K. Williams, B. Bräutigam, S. Saul, H. Williams, K. Bowen, and C. L. Giles. Bbookx: Design of an automated web-based recommender system for the creation of open learning content. In J. Bourdeau, J. Hendler, R. Nkambou, I. Horrocks, and B. Y. Zhao, editors, *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, Companion Volume*, pages 929–933. ACM, 2016.
- [37] J. Rihák and R. Pelánek. Measuring similarity of educational items using data on learners’ performance. In X. Hu, T. Barnes, A. Hershkovitz, and L. Paquette, editors, *Proceedings of the 10th International Conference on Educational Data Mining, EDM 2017, Wuhan, Hubei, China*. International Educational Data Mining Society (IEDMS), 2017.
- [38] S. Ritter, J. R. Anderson, K. R. Koedinger, and A. Corbett. Cognitive tutor: Applied research in mathematics education. *Psychonomic bulletin & review*, 14(2):249–255, 2007.
- [39] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.
- [40] K. Thaker, Y. Huang, P. Brusilovsky, and H. Daqing. Dynamic knowledge modeling with heterogeneous activities for adaptive textbooks. In K. E. Boyer and M. Yudelson, editors, *Proceedings of the 11th International Conference on Educational Data Mining, EDM 2018, Buffalo, NY, USA*. International Educational Data Mining Society (IEDMS), 2018.
- [41] D. Tudhope and C. Taylor. Navigation via similarity: Automatic linking based on semantic closeness. *Information Processing & Management*, 33(2):233–242, 1997.
- [42] M. Wang, H. Chau, K. Thaker, P. Brusilovsky, and D. He. Concept annotation for intelligent textbooks. *CoRR*, abs/2005.11422, 2020.
- [43] G. Weber and P. Brusilovsky. ELM-ART: An adaptive versatile system for web-based instruction. *International Journal of Artificial Intelligence in Education*, 12(4):351–384, 2001.

Image Reconstruction of Tablet Front Camera Recordings in Educational Settings

Rafael Wampfler
Dept. of Computer Science
ETH Zurich, Switzerland
wrafael@inf.ethz.ch

Andreas Emch
Fision AG, Switzerland
ae@fision-
technologies.ch

Barbara Solenthaler
Dept. of Computer Science
ETH Zurich, Switzerland
solenthaler@inf.ethz.ch

Markus Gross
Dept. of Computer Science
ETH Zurich, Switzerland
grossm@inf.ethz.ch

ABSTRACT

Front camera data from tablets used in educational settings offer valuable clues to student behavior, attention, and affective state. Due to the camera's angle of view, the face of the student is partially occluded and skewed. This hinders the ability of experts to adequately capture the learning process and student states. In this paper, we present a pipeline and techniques for image reconstruction of front camera recordings. Our setting consists of a cheap and unobtrusive mirror construction to improve the visibility of the face. We then process the image and use neural inpainting to reconstruct missing data in the recordings. We demonstrate the applicability of our setting and processing pipeline on affective state prediction based on front camera recordings (i.e., action units, eye gaze, eye blinks, and movement) during math-solving tasks (active) and emotional stimuli from pictures (passive) shown on a tablet. We show that our setup provides comparable performance for affective state prediction to recordings taken with an external and more obtrusive GoPro camera.

Keywords

Front Camera Setup, Inpainting, Affective Computing, Classification, Deep Learning

1. INTRODUCTION

Tablet computers have found quick application in education [14] as the technology offers new opportunities to students and teachers. It has been shown that tablets can influence learning pathways [19] and improve digital skills [47]. Moreover, tablets typically have built-in cameras, which can be used to unobtrusively record the student during the learning. Such data offers valuable clues to experts about the student's learning behavior and attention. Student observation has been implemented in studies with external camera setups [56]. Such frontal-view camera data can also be used for predictions of the affective states of a student based on

facial feature extraction [46], which works robustly even with low-resolution recordings [43]. Affective states are psychological constructs describing emotions (short-term) and moods (long-term) elicited by a stimulus [36, 51], and their impact on learning has attracted considerable attention in research on intelligent tutoring systems and education [3, 13, 41]. For example, Craig et al. [12] have found a positive correlation between learning and flow and a negative correlation between learning and boredom.

Using external cameras for frontal view recordings of students provides an optimal viewing angle for robust facial feature extraction and affective state prediction. However, such setups require externally positioned cameras, which can be obtrusive and further depend on timestamp synchronization with the digital learning environment. Using tablet computers for learning circumvents these problems, as the built-in camera can be leveraged and timestamps are inherently in sync. Built-in cameras have, however, a sub-optimal viewing angle, leading to partially occluded and skewed faces in the recordings that makes it difficult to robustly extract facial features for affect prediction.

In this paper, we therefore propose a camera setup for tablet computers and a deep learning-based image processing pipeline to reconstruct high-quality facial recordings of students. The setup requires a small mirror to be attached to the camera to improve the visibility of the face. Then, the image is reconstructed using a neural inpainting approach. We demonstrate the advantage of this setup and our reconstruction by an application for predicting affective states. The high quality of the reconstructed image enables facial feature extraction, such as head pose, eye gaze, and facial landmarks. We compare our method with an external camera setup (GoPro camera) and show that we can achieve a similar performance for predicting two levels (high and low) of valence and arousal for students performing active tasks, i.e., solving math tasks (up to 0.73 AUC) and students performing passive tasks, i.e., exposed to emotional stimuli from pictures (up to 0.80 AUC).

2. RELATED WORK

Inpainting. Image inpainting is an image processing method to reconstruct missing or corrupted regions of an image. Common application areas include image restoration (e.g.,

Rafael Wampfler, Andreas Emch, Barbara Solenthaler and Markus Gross "Image Reconstruction of Tablet Front Camera Recordings in Educational Settings" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 245 - 256

removing scratches and text) [34], photo-editing (e.g., object removal) [50], and image coding and transmission (e.g., recovering the missing blocks) [54]. In this work, we focus on the specific task of face completion. Popular non-learning based approaches applied to faces consist of patch-based methods, where image patches are copied to missing areas. Similar patches can be identified by using a face image dataset [58]. We refer to Guillemot and Le Meur [21] for a complete overview of non-learning based models.

While non-learning based methods can have difficulties to ensure consistent image structures [24, 45, 55], learning-based approaches typically generate smoother results. A popular line of learning-based methods uses generative adversarial networks (GAN) to inpaint missing regions of an image. GANs consist of a generative network to create a new image and a discriminator network to distinguish the new image from actual ground truth images. Using such a GAN approach, Malesevic et al. [37] reported a peak signal-to-noise ratio (PSNR) of up to 20.57 for inpainting missing regions in faces. A similar performance of up to 20.2 PSNR and 0.84 structural similarity (SSIM) was achieved by Li et al. [31] using an encoder-decoder network as the generator, a local and global loss function and a semantic regularization term. On the other hand, Liao et al. [32] used a collaborative model by training a GAN simultaneously on multiple tasks (i.e., face completion, landmark detection, and semantic segmentation). Using this knowledge-sharing approach, they reported a PSNR of up to 31.5 and an SSIM of 0.97 on face inpainting.

Convolutional neural networks (CNN) have been used for image inpainting as well. The encoder compresses the image with convolutional operations into a latent space, and the decoder reconstructs the image from the compressed representation. Guo et al. [22] proposed an encoder-decoder network using full-resolution residual blocks. For face inpainting, they reported a PSNR of 29 and an SSIM of 0.95. On the other hand, Liu et al. [35] achieved a PSNR of 34.69 and an SSIM of 0.99 by adding a coherent semantic attention layer to the encoder. One disadvantage of this method is its long runtime of 0.82 seconds per image of size 256×256 rendering this method inapplicable for real-time video processing with more than one frame per second. Another problem with existing CNN-based methods is that the convolution operations are applied both to the valid and missing pixels at the same time, which can lead to visual artifacts (e.g., color discrepancy and blurriness). To overcome this issue, Liu et al. [34] proposed partial convolutions, where the convolution operations are only applied to valid pixels by masking regions that need to be inpainted. The mask is updated during training of the network, including newly inpainted values. The authors demonstrated that the approach could produce semantically meaningful predictions also for inpainting regions with different shapes and sizes, achieving a PSNR of up to 34.34 and an SSIM of up to 0.95. We use this partial convolution approach to inpaint missing regions in images from front camera recordings. The dataset used for training the network is tailored to our use case.

Affective State Prediction. In our work, we focus on the prediction of affective states in the educational domain, such as in classroom settings and online courses. It was shown that affective states have an impact on learning gain in general,

and during math learning in particular [29, 44]. For example, Csikszentmihalyi [13] showed that engaged concentration has a positive effect on learning, while boredom negatively influences learning. Affective states are often grouped into basic emotions identified by Ekman [16] (i.e., anger, disgust, fear, happiness, sadness, and surprise) or described by the valence and arousal dimensions [40]. Valence indicates if an emotion is perceived as positive or negative, and arousal represents the intensity of an emotion.

Different modalities have been used to predict affective states using the valence-arousal space in educational settings. Acoustic features from student voices during interaction with tutors have been used to predict three levels of valence [33]. On the other hand, bio-sensors (i.e., skin conductance, heart rate, and skin temperature) and handwriting data have been successfully used to predict affective states in the valence-arousal space during math solving tasks [53]. Another line of research predicted valence and arousal using mouse and keyboard interaction data collected during text writing [49]. Multi-modal approaches fusing different modalities have also been introduced for the prediction of affective states. We refer to D’Mello et al. [15] that provides a concise overview of such methods.

Prediction of affective states from video recordings is one of the most popular approaches nowadays as it allows different features to be exploited, such as body language and posture, head movement, eye gaze and facial expressions [57]. Bosch et al. [6] calculated statistics (i.e., maximum, median and standard deviation) of the frame-level likelihood values of 19 different action units (AU) (i.e., facial muscle movements), the head position and gross body movement from webcam video recordings of students playing an educational physics game. They predicted two levels of boredom (0.61 AUC), confusion (0.65 AUC), delight (0.87 AUC), engagement (0.68 AUC) and frustration (0.63 AUC). Based on this work, Kai et al. [26] found that an interaction-based model using timing and counting-based features performs worse than the video-based model. Similarly, using a math tutor, Arroyo et al. [2] found facial expressions to be more predictive for confidence, frustration, excitement, and interest than conductance bracelets, pressure mice, and a posture analysis seat. Also in other domains facial expressions have found to be a good predictor for affective states. In text comprehension tasks, confusion (0.64 AUC), engagement (0.55 AUC), and frustration (0.61 AUC) have been successfully predicted using 20 different AUs [11]. On the other hand, Grafsgaard et al. [20] found upper face movements predictive for engagement, frustration, and learning in a setting consisting of a programming tutor and a webcam. Finally, based on eye gaze features (e.g., fixation and view angle) extracted from a specialized eye capturing device, boredom (69%) and curiosity (73%) have been successfully predicted on two levels each [25]. A survey of different video-based approaches for predicting affective states is provided by Zeng et al. [57].

A majority of the existing vision-based approaches use external devices, such as webcams, and rely on posed facial expressions to predict basic emotions [57]. In contrast, we present a novel setup for reliably recording the face of users based on the front camera of tablet computers only, and hence without the need for expensive devices or synchroniza-

tion between the devices. We demonstrate the usefulness of our setting by predicting affective states in terms of valence and arousal using data from an experiment containing spontaneous (non-posed) facial expressions. Finally, for our vision-based model, we fuse different existing approaches with novel features.

3. CAMERA SETUP

We present a low-cost hardware setup for recordings from the integrated front camera of a tablet computer, maximizing the visibility of the face of the users. Videos and images captured by the front camera are preprocessed, and missing parts are inpainted using a deep learning model to reconstruct the face of the users. Our approach is image-based and processes captured videos frame by frame.

3.1 Hardware Setup

While working on a tablet (e.g., writing with a stylus) it is convenient to have the device lying on the table (see Figure 1a). Due to the field of view of the front camera, only part of a users' face is visible. To adjust the field of view of the front camera, we attached a circular mirror (3cm radius) to the tablet using a hinge (see Figure 1b). The hinge was fixed with glue so that the mirror would remain in a stable position. The mirror was mounted with an angle of 75 degrees relative to the tablet. This angle was chosen so that the visibility of the face was maximized. Due to the mirror setup, the upper part of the recordings is mirror-inverted (see Figure 1c). Depending on the conditions of the illumination of the recording environment, the exposure time of the camera of the recording device (e.g., tablet) needs to be adapted accordingly so that the camera focuses on the face instead of the background. This adjustment of the exposure time can lead to an overexposed background (see Figure 1c).

3.2 Image Processing Pipeline

A raw image captured by the front camera is split by the mirror into two parts with the upper part of the image being mirror-inverted (see Figure 2A). To reconstruct the image, we propose a series of processing steps applied to the image (i.e., flattening the splitting boundary, face composition, image rotation, and extracting the face area). Image rotation and extraction of the face area are conducted as a preprocessing step for inpainting. Further, to train our inpainting model at a later stage, we assume that we have access to a dataset Ψ of square-shaped face images.

Splitting boundary. We apply a transformation to flatten the splitting boundary of the image (green line in Figure 2A), which simplifies image processing in the later stages and improves the final results qualitatively. We divide the image into 16 rectangles with equal width. An example of such a rectangle is shown in purple in Figure 2A. For each such rectangle, we transform the region defined by the vertices p_1, p_2, p_3 , and p_4 into the region defined by the vertices p_1, p_2, p_5 , and p_6 using a perspective transformation with linear interpolation. The location of these points can be calculated beforehand (or read from the image) because the mirror remains in a fixed position. The result of the transformation is shown in Figure 2B, where the splitting boundary (green) is a straight line.

Face composition. We rearrange the image by moving the part below the splitting boundary to the top and the flipped upper part to the bottom (see Figure 2C). The cut line defined by the mirror is shown in black. In addition, we adapt the height of this cut line because depending on the distance of the face, the missing part is increasing (increasing distance) or decreasing (decreasing distance). As a next step, we push the bottom corner of the upper face towards the middle by applying a second perspective transformation to the image so that the upper and lower part of the face are matching (see Figure 2D).

Image rotation. We then rotate the front camera image so that the eyes are horizontally aligned (see Figure 2E). Using dlib [28], we extract the coordinates of the facial landmarks belonging to the left and right eye. From these landmarks, we calculate the position of the center of each eye and rotate the image around the midpoint between the eye centers so that the line connecting the center of the eyes is horizontally aligned.

Face area. We extract the face area by computing a square bounding box encompassing the face (see the orange box in Figure 2E). This bounding box is defined by the vertices $p_7 = (x_7, y_7)$ and $p_8 = (x_8, y_8)$ and is given by

$$x_7 = c_{x,I} - \frac{w_{I_\Psi}}{2} * \frac{\delta_I}{\delta_{I_\Psi}} \quad (1)$$

$$x_8 = c_{x,I} + \frac{w_{I_\Psi}}{2} * \frac{\delta_I}{\delta_{I_\Psi}} \quad (2)$$

$$y_7 = c_{y,I} - \frac{c_{y,I_\Psi}}{h_{I_\Psi}} * (x_8 - x_7) \quad (3)$$

$$y_8 = c_{y,I} + \frac{h_I - c_{y,I_\Psi}}{h_{I_\Psi}} * (x_8 - x_7), \quad (4)$$

where I and I_Ψ denote an image of the front camera and an image in the dataset Ψ , respectively. The width and height in pixels of an image are given by w and h . The x - and y -coordinate of the midpoint between the left and right eye are denoted by c_x and c_y , respectively, and δ is the distance between the eyes. Here, we assume that the origin is located at the top left of the image.

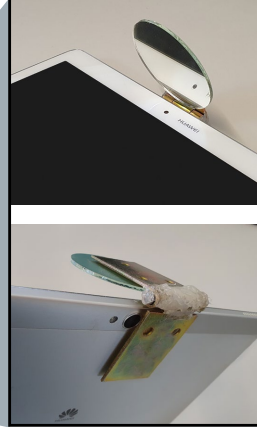
The part of the front camera image I outlined by the orange bounding box is then resized to the resolution $w_{I_\Psi} \times h_{I_\Psi}$ using bilinear interpolation. If the head of the user is close to the mirror, the face covers the full height of the image, and the bounding box might go over the upper and/or lower image borders. In such a case, we fill the parts overlapping the image with black pixels to get consistently sized bounding boxes (note that for visualization purpose only, the orange box in Figure 2E does not reflect this but instead is cut at the image border). We use the face detector of dlib [28] to test if a face and hence the landmarks of the eyes are identified in the image. In cases where the face cannot be detected, we use the landmarks of the eyes of the last image where the face could be identified (assuming that we have a video recording available, i.e., a series of images).

Inpainting missing area. As the last step in our image preprocessing pipeline, we inpaint the missing parts in the bounding box of the image (black region of the orange box in Figure 2E) with the neural inpainting approach of Liu

a) Experimental setup



b) Camera setup



c) Front camera recordings



Figure 1: The hardware setup. A user is working on the tablet (a). A mirror is attached to the tablet using a hinge (b). Due to the mirror reflections, the field of view of the front camera is changed so that the face of the participant is visible (c).

et al. [34] described in Section 3.3. We apply the neural inpainting only to the bounding box because it contains the important parts of the face (i.e., eyebrows, eyes, and mouth). We inpaint other parts of the image outside the bounding box using a simple Navier-Stokes based inpainting method provided by OpenCV [8] which is based on a circular neighborhood of three pixels for each inpainted pixel. Finally, we rotate the image back to its original orientation. This then leads to the final reconstructed image shown in Figure 2F.

3.3 Neural Inpainting

For the neural inpainting approach, we use the dataset Ψ of square-shaped face images with customized missing regions tailored to our application of tablet front camera recordings and then train the network on this dataset.

Training dataset. The model is trained on a large corpus of images from the dataset Ψ together with a mask for each image indicating the missing parts (a mask is a matrix with the same size as the image having a '1' entry for missing pixels and a '0' entry otherwise). We create the corresponding mask randomly and similar in shape (rectangle) to the expected mask in our front camera recordings (see Figure 3 for an example of two such masks applied to two images from the CelebA-HQ dataset [27]). Note that the mask (missing image region) is not necessarily horizontal but rotates if a user is rotating the tablet or the head (vertical in the extreme).

Inpainting method. Liu et al. [34] use a neural network that consists of an encoder E and a decoder D . The encoder network transforms the input image $\mathbf{I} \in \mathbb{R}^{M \times N}$ into a low-dimensional (latent) space $\mathbf{z} = E(\mathbf{I})$. The decoder then reconstructs the original image based on this low-dimensional representation $\hat{\mathbf{I}} = D(\mathbf{z})$. The encoder and decoder networks consist of $n = 8$ partial convolutional layers denoted as E_1, \dots, E_n and D_1, \dots, D_n for the encoder and decoder networks, respectively. Before each convolution operation, the image is constrained by the mask to condition the operation on only valid pixels. The mask is updated for the next layer removing masking for pixels where the convolutional operation operated on unmasked values. In addition, each layer in the

encoder network E_i is connected to the corresponding layer in the decoder network $D_i, \forall i \in \{1 \dots n\}$ using skip links. These skip links allow for copying unmasked pixels directly from the encoder to the decoder without passing the bottleneck (latent space). To direct the training of the network towards semantically meaningful inpaintings, a combination of four loss functions is used (i.e., per-pixel loss, perceptual loss, style loss, and total variation loss). Using these loss functions smooth transitions of the predicted masked values into their neighboring pixels is also taken into account. As activation functions Rectified Linear Unit (encoder) and a leaky version of a Rectified Linear Unit (decoder) are used.

4. AFFECTIVE STATE PREDICTION

We present the prediction of affective states as an example application of our mirror setup and image processing pipeline. Our classification pipeline can be generally applied to any recordings captured with a tablet front camera or an external camera (such as a GoPro). Our method assumes that we have access to reports of affective states of users based on the circumplex model of affect [48]. The circumplex model defines affective states in a two-dimensional space spanned by valence and arousal. The classification task then amounts to preprocessing the camera recordings to adjust the brightness and the frame rate and predicting valence and arousal based on features extracted from the adjusted camera recordings. Affectiva [39] provides out of the box predictions of the basic emotions and valence based on images and video recordings. However, initial tests revealed that these predictions are not of sufficient quality when applied to our use case. Thus, we developed our own set of features incorporating some additional features not taken into account by Affectiva, such as movement and fidgeting. Moreover, by using our own extracted features, we can predict arousal in addition to valence.

4.1 Preprocessing

First, we resample the camera recordings using FFmpeg [5] to a constant frame rate close to the mean frame rate. Depending on the recording device, the frame rate can vary (e.g., the frame rate can drop due to the higher load of the

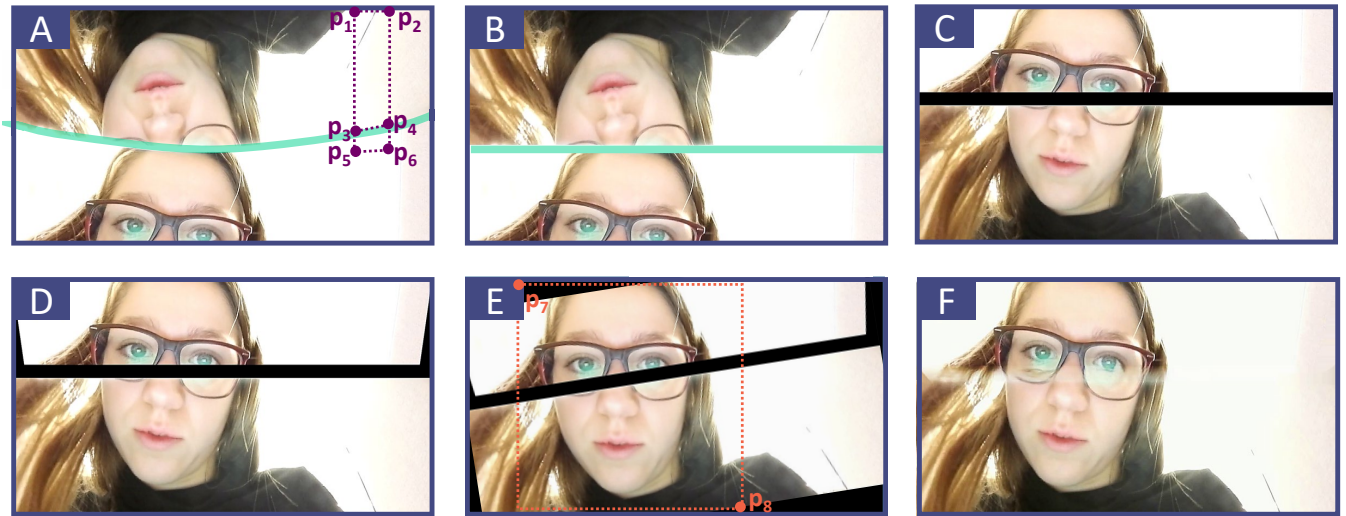


Figure 2: The main inpainting steps. The splitting boundary of front camera recordings (A) is flattened using a perspective transformation (B). The face is reconstructed from the upper and lower parts (C) and warped so that the upper and lower part match (D). Finally, after horizontally aligning the eyes (E), the missing regions (black) are inpainted (F).



Figure 3: Two example masks applied to images of the CelebA-HQ dataset [27].

device). A constant frame rate facilitates the extraction of the features and the processing of the recordings in later stages. In addition, we adjust the brightness of the recordings based on the brightness estimation of Affectiva [39] to improve the lighting of the face for the analysis. Depending on the conditions of illumination at recording time the face can be underexposed (too dark) or overexposed (too bright, e.g., when the camera is directed towards a lamp). This can hinder the accurate detection and extraction of facial features such as landmarks.

4.2 Feature Extraction

From the camera recordings, we extract several different feature types. We design all features such that they are independent of the frame rate (e.g., using percentages instead of absolute positions) to support cameras with different frame rates. To extract facial landmarks, eye gaze, and head position from the camera recordings, we rely on OpenFace [4] using static extraction (i.e., per frame without calibrating to a person). OpenFace also provides a confidence value $c(i) \in [0, 1]$ for each frame i indicating the confidence in the landmark detection estimate. If $c(i) < 0.82$, we discard the frames $i - 5, \dots, i + 5$ (i.e., 11 frames). The number of

frames to discard (11) and the threshold (0.82) were heuristically determined. All features are computed over a window containing N frames. If, after considering the confidence value, less than 80 % of the frames are remaining, we discard the window and the corresponding data point. Again, this threshold was determined heuristically. Where appropriate, we calculate for the different feature types basic statistics over the window (i.e., maximum, minimum, relative position of minimum and maximum, mean, standard deviation, and the slope of a fitted linear regression line), providing 282 features in total. In addition, to correct for differences between individuals related to facial expressions and posture, we normalize each feature according to a baseline by subtracting the feature calculated over a baseline period (e.g., watching a nature video putting the individuals in a relaxed state).

Action units. Facial action units (AUs) are based on the Facial Action Coding System (FACS) and identify independent motions of the face [17]. We extract basic statistics of the intensity (from 0 to 5) of 17 AUs covering motions in the eye, cheek, nose, mouth, and chin region. In addition, for each AU, we calculate the percentage of the presence (absent versus present) in the window. Moreover, the AUs can be directly mapped to the six basic emotions identified by Ekman [16]. Thus, for each basic emotion, we also calculate the basic statistics of the corresponding added up AUs.

Eye blinks. Researchers have found a correlation between eye blink frequency and stressful situations in a car driving simulation [23]. Similarly, a correlation between eye blinks and affective states in learning environments was found [38]. Here, we base the eye blink detection on the signal from the AU that represents eye closure as a continuous signal (from 0 to 5) with peaks indicating potential eye blinks. We detect peaks belonging to an eye blink by thresholding the signal according to the ratio between the prominence (how much a peak stands out measured as the vertical distance between the peak and its lowest contour line) and width of a

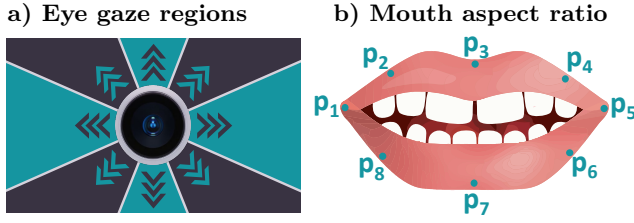


Figure 4: Eye gaze regions and mouth aspect ratio (MAR). The gaze angle is discretized into nine different gaze regions, including the center (gazing towards the camera lens) (a). MAR is calculated based on the height and width of the mouth (b).

peak. Heuristically, we found a threshold of 0.026 to provide the best results. We found that taking into account the width of the peaks is necessary to accurately detect peaks belonging to eye blinks because the prominence of the peaks differs among users and head pose. We extract the number of blinks and the basic statistics of the duration between blinks, the prominence, and the width of each blink. In addition, inspired by interbeat intervals (time interval between individual heartbeats) and the calculation of heartbeats thereof, we linearly interpolate the duration between two consecutive peaks surviving the threshold (i.e., eye blinks) to infer a continuous signal. We then calculate the number of eye blinks for every frame by taking the inverse of this interpolated signal. Subsequently, we again calculate the basic statistics over the number of eye blinks.

Eye gaze. The intention behind features related to eye gaze is that individuals might look away when thinking while solving math tasks or when looking at emotionally disturbing pictures. Thus, we compute the basic statistics on the angle in the x-direction (looking left-right) and y-direction (looking up-down) of the eye gaze averaged for both eyes and measured in radians in world coordinates. In addition, we discretize the eye gaze angle by defining nine different gaze regions (see Figure 4a). The center corresponds to a line of gaze directed towards the camera lens. For each of the nine regions, we count the number of occurrences and normalize it over $s * \text{fps}$, where s is the window size and fps is the frame rate per second (so that it is independent of the used camera, i.e., the frame rate).

Mouth aspect ratio. Previously, the mouth aspect ratio (MAR) was used to detect driver drowsiness [52]. It is defined by the ratio between the height and the width of the mouth, which is increased when opening the mouth (see Figure 4b):

$$\text{MAR} = \frac{\|p_2 - p_8\| + \|p_3 - p_7\| + \|p_4 - p_6\|}{3 * \|p_5 - p_1\|}. \quad (5)$$

Each point $p_i, \forall i \in \{1, \dots, 8\}$, is defined as the average of the inner and outer mouth landmarks. From the MAR, we calculate the basic statistics.

Head Movement. From the longest head moving sequence of an individual in the window, we extract the position of the first frame of the sequence in relation to the beginning of the window, the duration of the movement, and the total distance of the movement. The position of the first frame and the duration are normalized by $s * \text{fps}$. We also sum up the total

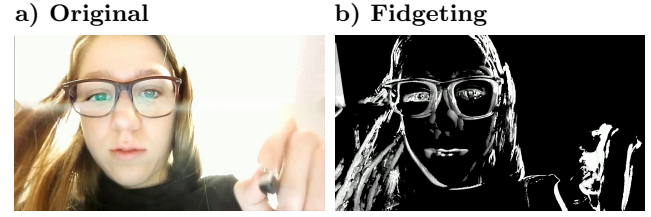


Figure 5: Fidgeting of a user. From the original image (a), the fidgeting image (b) is calculated by pixel-wise thresholding the difference of the current (a) to the past grayscale images.

distance moved over the entire window to capture individuals continually moving back and forth. In addition, we calculate the basic statistics of the velocity and acceleration of the head movements in the window. All these features are extracted for the x-axis, y-axis, and z-axis separately. Finally, we also extract the basic statistics of the distance of the head to the camera in the three-dimensional space.

Fidgeting. Navarathna et al. [42] introduced a fidgeting index for predicting movie ratings from audience behavior by calculating the total energy individuals are using for the movement. In contrast to features related to the head movement, fidgeting captures all the movement in the video (i.e., also body and face). First, we define the grayscale adaptive background b_{gray} , which is a weighted average of past frames. To calculate the energy E for a new frame f_{gray} (converted into grayscale), we subtract the adaptive background b_{gray} from f_{gray} , binarize the image by thresholding it, and then calculating the percentage of surviving pixels with respect to the camera resolution (see Figure 5b). We have chosen the threshold such that noise from the background is minimized, and the visibility of movements is maximized. Finally, the adaptive background is updated using

$$b_{\text{gray}} = (1 - a) * b_{\text{gray}} + a * f_{\text{gray}}, \quad (6)$$

where a is a weight term (we found $a = 0.2$ to provide the qualitatively best results). From the energy E of each frame in the window, we calculate basic statistics, sum up the energies over all frames and use the position of the frame with minimum and maximum energy normalized by $s * \text{fps}$.

4.3 Classification

We build the ground truth for our classifiers by splitting valence and arousal into two levels (high and low). We then use classifiers to predict these levels based on the features extracted from the camera recordings. In addition, we remove features having a correlation greater than a threshold, select features based on the ANOVA F-value between the class labels and the features, and standardize the features to have zero mean and unit variance. We use four different classifiers (i.e., Random Forest, Support Vector Machine, k-Nearest Neighbors and Gaussian Naive Bayes) because these classifiers have been most promising in initial tests and they have shown to provide good results for predicting affective states from video data in other works [25, 10, 6]. We use leave-one-user-out cross-validation to evaluate our models, which ensures that data of a participant is not used for training and testing at the same time. Finally, we optimize the hyperparameters (i.e., number of selected features, the threshold for

removing correlated features, and parameters of the model) using random search with nested cross-validation.

5. RESULTS

We conducted a qualitative and quantitative evaluation of our mirror setup and image processing pipeline with neural inpainting and investigated the applicability of our setup to predict affective states during math-solving tasks (active) and during exposure to emotional stimuli from images (passive). For training the neural inpainting model, we have used the celebA-HQ dataset [27] consisting of 30000 face aligned colored images from celebrities with a resolution of 1024×1024 pixels (we downsampled the images to 512×512 pixels). We split the dataset into a training set of 25000 images, a test set of 2500 images and a validation set of 2500 images. We set the parameters for the network in the same way as proposed by Liu et al. [34]. The results of the affective state prediction are based on a Random Forest classifier since this was the best performing model. Hyperparameters were optimized using random search with 50 iterations. Finally, for measuring the performance of our model, we used the area under curve (AUC) of the receiver operating characteristic curve and accuracy (chance level = 0.5).

5.1 Experiment

We reused a dataset that we collected in a controlled lab experiment [53]. The dataset consists of data from 88 participants (45 female) from age 18 to 29 (mean = 22.1, standard deviation SD = 2.0) of university students in the bachelor program. The participants used a Huawei MediaPad M2 10.0 tablet running Android 5.1 during the experiment. They were recorded by the front camera (resolution of 1280×720 pixels) using our proposed mirror construction setup and a GoPro HERO3 camera (frame rate per second FPS of 59.94 and a resolution of 1920×1080 pixels) (see the setup in Figure 1a). Due to the varying load of the tablet during the experiment, the fps was variable (mean = 20.02, SD = 1.92). We resampled the recordings from the tablet and the GoPro to an fps of 25 and 60, respectively. To synchronize the timestamps between the GoPro and the tablet, a beep signal was played on the tablet before the start of each session.

The study procedure consisted of three main steps conducted on the tablet to collect baseline data and trigger different affective states. First, each participant was watching a seven minutes nature video, which served as a baseline. Second, the participants were presented 40 pictures in random order from the International Affective Picture System (IAPS) [30] for around 20 minutes. The IAPS is a collection of 1182 pictures standardized in terms of valence and arousal and is widely used in psychological research for the study of emotions. Each image was shown for ten seconds and was followed by a ten seconds fixation cross. The 40 images have been selected from the IAPS dataset such that a wide range of the valence-arousal space was covered.

Finally, each participant solved multiple-choice math tasks for approximately 30 minutes. The math tasks were selected from a collection of math tasks provided by ACT [1] and divided into three different conditions varying in difficulty level, available completion time, and monetary reward (participants were rewarded and penalized depending on the

correctness of the solution and started with a credit of CHF 40). In the repetitive condition, easy and repetitive (i.e., similar) tasks were presented with more than enough available time to solve the tasks and a minor reward (+CHF 0.2) and penalty (−CHF 0.2). In the challenge condition, tasks with medium difficulty levels were shown with sufficient time to solve the tasks, and a large monetary reward (+CHF 2) but an only minor penalty (−CHF 0.2). The overchallenge condition consisted of tasks with a high difficulty level, insufficient time to solve the tasks and a small monetary reward (+CHF 0.2) but a large penalty (−CHF 2). The tasks were presented in six blocks. Each block contained tasks from a specific condition, and each condition was assigned randomly to two blocks.

After each image and math task, participants were asked to fill in the self-assessment manikin (SAM) [7] to judge their current valence and arousal level on a 9-point Likert scale. To build our affective prediction model, we split the valence and arousal ratings of the participants into two classes (low $\in \{1, \dots, 3\}$ and high $\in \{7, \dots, 9\}$). For IAPS, the number of data points amounted to 843 (1206) and 1218 (982) for low and high valence (arousal), respectively. On the other hand, for math tasks, the number of low and high valence (arousal) ratings amounted to 724 (1380) and 1422 (726), respectively.

5.2 Face Recognition

We provide qualitative and quantitative results of our setup using neural inpainting. In particular, we compare our results to recordings taken by the GoPro camera.

Qualitative evaluation. Figure 6 shows the facial landmarks detected by OpenFace for three participants from the front camera without inpainting, using neural inpainting, and from the GoPro. The positions of the detected landmarks without inpainting are inferior compared to neural inpainting. For participant 3, the landmarks at the upper face (eyebrows, eyes, and nose) are misaligned without inpainting. Often no facial landmarks could be detected (see Figure 6 participants 1a and 2a). With our neural inpainting approach, we achieved a qualitatively good recovered image independent of the position of the missing region (e.g., eyes and mouth). It is noteworthy that the inpainting and facial landmark detection also worked for participants wearing glasses. The detected landmarks after neural inpainting are similar to the landmarks detected from the GoPro recordings (see Figure 6c). Depending on the position of the head, the landmarks of the eyes and the mouth can become locally condensed in the GoPro recordings, and it might be hard to distinguish slight facial movements. On the other hand, from the front camera, the recordings are frontal, and the variations of facial parts (e.g., eye and mouth) are better visible.

Quantitative evaluation. Table 1 presents the average confidence in landmark detection of OpenFace over all frames for the IAPS and math-solving tasks and the full recordings (including also parts not belonging to the IAPS and math tasks). Reported confidence values by OpenFace are between 0 (not confident) and 1 (fully confident). Without inpainting, the confidence values are low, and standard deviations are high due to the imperfect recognition of landmarks. Without inpainting landmarks were often only detected correctly

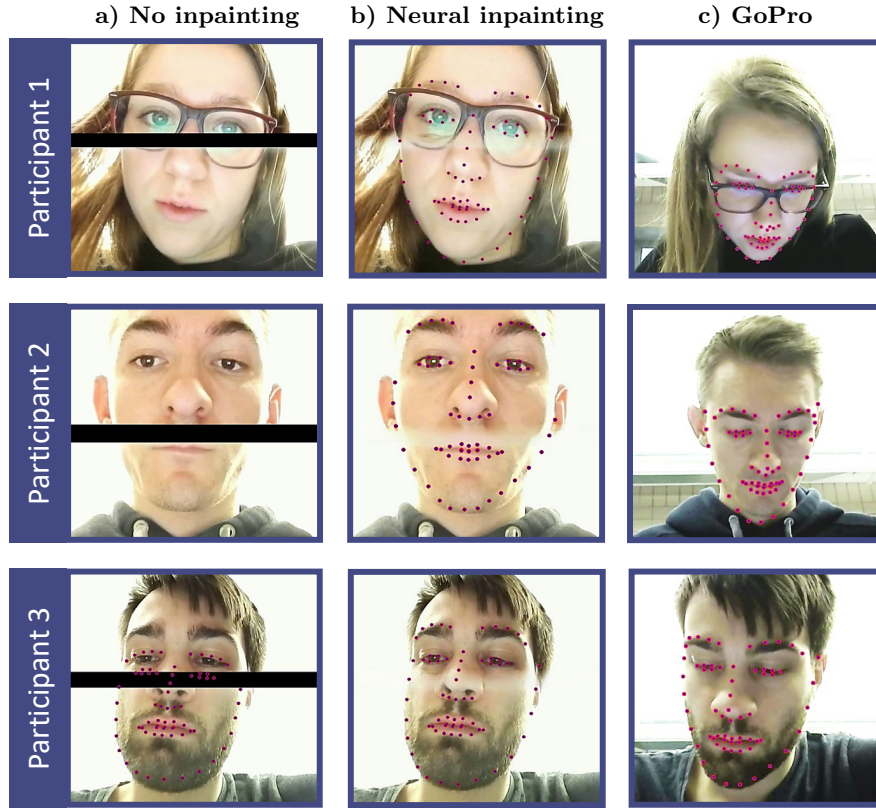


Figure 6: Recordings of three participants. The facial landmarks were detected from the front camera recordings without inpainting (a) and with neural inpainting (b) and from the external GoPro camera (c). If no landmarks are visible, no landmarks were detected by OpenFace.

Table 1: Means of framewise confidence in landmark detection for different camera sources, tasks (math and IAPS) and the full recordings. Confidence values range from 0 (not confident) to 1 (fully confident). Standard deviations are given in brackets.

Source	IAPS	Math	Complete
Front (no inpainting)	0.79 (0.36)	0.48 (0.45)	0.68 (0.42)
Front (inpainting)	0.94 (0.14)	0.90 (0.22)	0.93 (0.18)
GoPro	0.97 (0.08))	0.93 (0.17)	0.95 (0.12)

when the missing regions were situated above the eyebrows (i.e., no landmarks have been affected). After applying neural inpainting, the confidence values increased by 19 % and 88 % during IAPS and math sequences, respectively. When considering the full video recordings, the increase amounts to 37 %. In addition, the standard deviation decreased substantially. This increase of the confidence leads to an increase in the number of samples (if a window used during feature extraction contained less than 80 % frames with a confidence value above 0.82 we discarded the corresponding data point). For IAPS, this lead to 348 and 383 additional samples for valence and arousal, respectively. For the math tasks, this amounted to 1233 and 1179 additional samples for valence and arousal, respectively. Finally, the confidence in landmark detection of the GoPro recordings is comparable to the front camera recordings with neural inpainting. In general, for recordings taken during exposure to a stimulus set of images

Table 2: Performance of Random Forest on the math and IAPS data from two levels (low and high) of valence and arousal based on the front camera recordings with neural inpainting and the GoPro recordings. The chance level for accuracy and AUC is 0.5.

Source	Data	AUC	Accuracy
Front camera	Math (valence)	0.73	68 %
	Math (arousal)	0.54	57 %
	IAPS (valence)	0.80	73 %
	IAPS (arousal)	0.70	66 %
GoPro	Math (valence)	0.76	72 %
	Math (arousal)	0.58	62 %
	IAPS (valence)	0.78	72 %
	IAPS (arousal)	0.73	67 %

the mean confidence is higher than during math tasks. This can be attributed to the fact that while solving math tasks, participants were moving more, which leads more often to suboptimal head positions for landmark detection. This finding is also reflected in the higher standard deviations of the confidence values for math tasks.

5.3 Classification Performance

Before predicting the affective states, the reconstructed front camera recordings and the GoPro recordings were preprocessed (see Section 4.1). Features were extracted using a ten seconds window encompassing the on-screen time of each pic-

Table 3: Number of occurrences of each feature type in the ten most predictive features. The numbers are provided for each of the four models (MV = math valence, MA = math arousal, IV = IAPS valence, IA = IAPS arousal).

Feature Type	MV	MA	IV	IA
Action units	0	2	2	3
Eye blinks	1	4	0	1
Eye gaze	1	2	2	0
Mouth aspect ratio	0	0	0	0
Head Movement	5	2	5	6
Fidgeting	3	0	1	0

ture and the last ten seconds of each math task because each picture was presented for ten seconds and the minimum task duration was ten seconds (see Section 4.2). Table 2 presents the performance of our model for predicting two levels (low and high) of valence and arousal. Based on the findings that the confidence in landmark detection increased up to 88% with neural inpainting, we used only the front camera recordings with neural inpainting. Using these recordings, our model achieved a performance of 0.73 AUC and 0.80 AUC for predicting valence on math tasks and IAPS, respectively. For predicting arousal, the performance drops and is only at random level for math tasks (0.54 AUC), while for IAPS it is above random (0.70 AUC). A similar pattern is visible for the GoPro recordings. While for predicting arousal based on the math tasks, the performance is close to random (0.58 AUC), all other predictions are above random. In summary, the predictions using the front camera are comparable to using the GoPro recordings with a maximum difference of 0.04 AUC. For predicting valence based on IAPS, the performance from the front camera recordings (0.80 AUC) exceeds the performance achieved by using the GoPro (0.78 AUC).

Feature importance. Table 3 presents the number of occurrences of each feature type in the ten most important features for each of the four models. We analyzed the feature importance using the Gini importance measure provided by the Random Forest classifier. Features related to head movement contributed the most for predicting valence based on math tasks (five features) and valence and arousal based on IAPS (five and six features). For predicting arousal based on math tasks, eye blinks provided four out of the ten most important features. There were no MAR features among the top ten features for any model. However, all feature types appeared in the top 30 ranked features of each model. For the model based on the math tasks, the maximum moved distance in the x-direction and the number of eye blinks were the highest scoring features for predicting valence and arousal, respectively. For the model based on IAPS, the mean acceleration in the x-direction and mean velocity in the x-direction were most important for predicting valence and arousal, respectively. Interestingly, head movement along the x-axis (left and right) was more informative than along the z-axis (forward and backward).

5.4 Runtime

We conducted a runtime analysis of the different parts of our inpainting pipeline and affective state prediction model. Our computing environment consisted of an Intel® Core™ CPU

i9-9900K @ 3.60GHz and an NVIDIA GeForce® RTX 2080 Ti. Processing one frame consisted of flattening the splitting boundary, face composition, image rotation and extracting the face area (mean = 17.07 ms, SD = 4.74 ms), detecting the position of the eyes using dlib (mean = 74.66 ms, SD = 6.43 ms), using the deep learning model to inpaint missing regions in the face (mean = 76.25 ms, SD = 13.81 ms) and inpainting the background of the image (mean = 47.01 ms, SD = 11.87 ms). Summing up these values leads to a processing time for one frame of 214.99 ms. Prediction of a new data point consisted of feature extraction (mean = 16.37 ms, SD = 2.18 ms) and using the Random Forest classifier for predicting valence and arousal (mean = 6.43 ms, SD = 10.52 ms), leading to a total prediction time of 22.8 ms.

6. DISCUSSION

Our findings show that it is possible to use our tablet-based front camera setup and processing pipeline to accurately capture users for extracting features such as facial landmarks and movement of the head and body. Our neural inpainting pipeline provides a qualitatively accurate restoration of missing regions caused by our mirror construction setup and increases the confidence in landmark detection by up to 88%. Compared to recordings from a GoPro camera, our setup provides better results in terms of face visibility (frontal view). Thus, it potentially facilitates the recognition of minor facial movements (e.g., mouth and eyes). In particular, for solving math tasks we found the recording conditions of the GoPro more challenging due to the viewing angle (participants were bending over the tablet). This resulted in lower confidence in landmark detection (0.93 for math tasks versus 0.97 for IAPS). Similarly, the front camera recordings with neural inpainting showed higher confidence in landmark detection during IAPS (0.94) compared to solving math tasks (0.90). During the exposure to a stimulus set of images from the IAPS dataset, participants were sitting straight, implicating that the splitting boundary was located at the forehead, which made inpainting easier. In contrast, during solving math tasks, the splitting boundary was often located in the middle (eye) or lower part of the face (mouth), creating a more challenging situation for our neural inpainting model.

We showed the applicability of our setup for predicting affective states during active (math-solving) and passive (exposure to pictures) tasks based on the recordings from the front camera. Our model achieved better performance on IAPS (up to 0.80 AUC) than on the math tasks (up to 0.73 AUC). Due to the active involvement of the participants while solving math tasks, participants were moving more, which made accurate tracking of facial landmarks, AUs, and eye gaze more demanding. In addition, our model performed better for predicting valence (0.73 AUC and 0.80 AUC) than arousal (0.54 AUC and 0.70 AUC). One-third of the participants rated arousal constantly as low or high without showing much variation. This finding can affect the generalization of our model to other participants for predicting arousal. In addition, although affective states are universal, they also have components that are individual to a person [18]. This makes it harder to predict an affective state of a person without having training data available of that person. Comparing the performance of our affective prediction pipeline to other research is difficult because most existing work [10, 57] predicted basic emotions and used other settings.

Our analysis of the feature importance showed that head movement is a predictive feature in contrast to MAR. Some AUs capture movements of the mouth. Thus, we analyzed the correlation between MAR and AUs specific to the mouth region. The correlations between the MAR feature and the AUs specifying lip corner puller (-0.15 , $p\text{-value} = 0.15$), opening the mouth (0.25 , $p\text{-value} = 0.13$) and jaw drop (0.045 , $p\text{-value} = 0.26$) have all been low and not significant.

In comparison to recordings from the GoPro, our model based on front camera recordings performed equally well and even better for predicting valence on IAPS (0.80 AUC versus 0.78 AUC). This renders our setup a viable alternative to more expensive equipment such as a GoPro. Our setup comes at low costs (CHF 5), is unobtrusive, can easily be mounted, is flexible in the application (e.g., in classrooms or at home), and eliminates the need for synchronizing different devices. In contrast to external cameras, the camera (i.e., the lens) in our setup is small and unobtrusive. Some participants reported after the experiment that they got slightly distracted by the GoPro but not by our mirror setup. Similarly, in the video recordings, we recognized that participants were sometimes glancing at the GoPro. Finally, with a processing time of 214.99 ms per frame, our pipeline can handle four frames per second. Our affective prediction pipeline is capable of making 43 new predictions every second.

Limitations. We acknowledge potential limitations to our approach presented in this paper. Our setup is constrained by the lighting conditions, head pose, and occlusions from hand movement. We believe that other camera setups suffer from the same constraints. Further, our mirror construction is a prototype and not yet ready for production. Although during the experiment the construction proved to be stable, it can be improved in terms of stability and flexibility. Neural inpainting provided qualitatively satisfactory results for most facial parts. However, if the splitting boundary is covering the eyes (i.e., both eyes are occluded), it is hard for the inpainting model to reconstruct the eyes at a qualitatively high level. Consequently, the landmark detection cannot recover eye gaze and eye blinks, but still detects other facial features. In addition, although the CelebA-HQ dataset consists of facial images from celebrities with diverse ethnicity, age and facial characteristics (e.g., glasses and facial hair), our inpainting method might be less appropriate for students who are underrepresented in the CelebA-HQ dataset. We further acknowledge that our experiment is restricted to math tasks and exposure to emotional stimuli from pictures in a lab environment with bachelor students. We are optimistic that our approach generalizes to a broader population and to other tasks given that we used active (math-solving) and passive (exposure to pictures) tasks and assuming a proper baseline normalization of the features. Finally, we have predicted valence and arousal on two levels omitting data points in the medium range (4 to 6). Our main contribution is the novel mirror construction and the processing pipeline. We have mainly built our affective prediction model for demonstrating the applicability of our setup. Nevertheless, we believe that our features and pipeline can be interesting for other researchers predicting affective states based on video data.

Future work. Future research comprises refining and extending our hardware setup and inpainting pipeline, as well as evaluating our affective prediction model in other domains. In particular, realtime performance would be desirable for on-the-spot assessment of a student's affective state. The CelebA-HQ dataset, which we used to train our inpainting model, contains only images with a frontal view of faces. In our recordings, individuals are captured at different angles. Thus, rotation of the recordings or using a dataset providing faces at different angles can improve the neural inpainting model. In addition, a deep learning model could be trained on our features for affective prediction, and the feature set could be extended by gesture-based features. Such features have shown to be promising for predicting affective states [9].

7. CONCLUSION

In this paper, we presented a hardware setup consisting of a cheap and unobtrusive mirror construction to improve the visibility of the face in tablet-based front camera recordings. Recordings were processed using an inpainting pipeline consisting of a neural network for reconstructing missing data in the recordings. We showed that the mirror construction improved the visibility of the face in situations where external cameras (e.g., GoPro) struggle. With a qualitative and quantitative evaluation, we demonstrated that we could achieve results comparable to a GoPro camera. In particular, neural inpainting improved confidence in facial landmark detection by up to 88% . We showed the applicability of our setup and processing pipeline on affective state prediction based on front camera recordings. Our model consisted of features capturing information from movement, eyes, and face. We evaluated our affective prediction model on data from a lab experiment with 88 participants using leave-one-user-out cross-validation. Participants were solving math tasks (active) and were exposed to emotional stimuli from pictures (passive). Our model accurately predicted two levels (low and high) of valence (up to 0.80 AUC) and arousal (up to 0.73 AUC) using data from the front camera. These results were comparable to results obtained using recordings from a GoPro camera (up to 0.78 AUC for valence and up to 0.73 AUC for arousal). The novelty of our contribution consists of the hardware setup and processing pipeline. In addition, we proposed features for affective state prediction, which can be useful for other researchers. Our setup is cheap (CHF 5), easy to mount, and can be used in classrooms or at home. Besides affective state prediction, it can be used to monitor students or analyzing attention. Most existing approaches use external cameras such as GoPros or webcams, which are more expensive, more difficult to handle, and are exposed to time synchronization problems. In our setup, the camera data is recorded on the same device as the task is conducted, and thus we circumvent such time synchronization issues in an elegant way. The findings of this work are important because they support the emerging trend of using tablet computers in the classroom and for learning at home by simplifying student recording and assessment.

Acknowledgments. We thank Katja Wolff and Fraser Rothnie for their assistance in creating the figures.

8. REFERENCES

- [1] ACT. The act technical manual, 2017.
- [2] I. Arroyo, D. G. Cooper, W. Bursleson, B. P. Woolf, K. Muldner, and R. Christopherson. Emotion sensors go to school. In *AIED*, volume 200, pages 17–24. Citeseer, 2009.
- [3] R. S. J. d. Baker, S. M. Gowda, M. Wixon, J. Kalka, A. Z. Wagner, A. Salvi, V. Aleven, G. W. Kusbit, J. Ocumpaugh, and L. Rossi. Towards sensor-free affect detection in cognitive tutor algebra. *International Educational Data Mining Society*, 2012.
- [4] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 59–66. IEEE, 2018.
- [5] F. Bellard. Ffmpeg. <https://ffmpeg.org/>.
- [6] N. Bosch, S. D’Mello, R. Baker, J. Ocumpaugh, V. Shute, M. Ventura, L. Wang, and W. Zhao. Automatic detection of learning-centered affective states in the wild. In *Proceedings of the 20th international conference on intelligent user interfaces*, pages 379–388, 2015.
- [7] M. M. Bradley and P. J. Lang. Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry*, 25(1):49–59, 1994.
- [8] G. Bradski. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*, 2000.
- [9] D. M. Bustos, G. L. Chua, R. T. Cruz, J. M. Santos, and M. T. Suarez. Gesture-based affect modeling for intelligent tutoring systems. In *International Conference on Artificial Intelligence in Education*, pages 426–428. Springer, 2011.
- [10] R. A. Calvo and S. D’Mello. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on affective computing*, 1(1):18–37, 2010.
- [11] Y. Chen, N. Bosch, and S. D’Mello. Video-based affect detection in noninteractive learning environments. *International Educational Data Mining Society*, 2015.
- [12] S. Craig, A. Graesser, J. Sullins, and B. Gholson. Affect and learning: an exploratory look into the role of affect in learning with autotutor. *Journal of educational media*, 29(3):241–250, 2004.
- [13] M. Csikszentmihalyi. *Flow: The psychology of optimal experience*. New York: Harper & Row, 1990.
- [14] C. Ditzler, E. Hong, and N. Strudler. How tablets are utilized in the classroom. *Journal of Research on Technology in Education*, 48(3):181–193, 2016.
- [15] S. K. D’Mello, N. Bosch, and H. Chen. *Multimodal-Multisensor Affect Detection*, page 167–202. Association for Computing Machinery and Morgan & Claypool, 2018.
- [16] P. Ekman. Basic emotions. *Handbook of cognition and emotion*, 98(45–60):16, 1999.
- [17] P. Ekman and W. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement: Investigator’s Guide 2 Part*. Consulting Psychologists Press, 1978.
- [18] H. A. Elfenbein and N. Ambady. Universals and cultural differences in recognizing emotions. *Current directions in psychological science*, 12(5):159–164, 2003.
- [19] G. Falloon. Young students using iPads: App design and content influences on their learning pathways. *Computers & Education*, 68:505–521, 2013.
- [20] J. Grafsgaard, J. B. Wiggins, K. E. Boyer, E. N. Wiebe, and J. Lester. Automatically recognizing facial expression: Predicting engagement and frustration. In *Educational Data Mining 2013*, 2013.
- [21] C. Guillemot and O. Le Meur. Image inpainting: Overview and recent advances. *IEEE signal processing magazine*, 31(1):127–144, 2013.
- [22] Z. Guo, Z. Chen, T. Yu, J. Chen, and S. Liu. Progressive image inpainting with full-resolution residual network. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2496–2504, 2019.
- [23] M. Haak, S. Bos, S. Panic, and L. J. M. Rothkrantz. Detecting stress using eye blinks and brain activity from EEG signals. *Proceeding of the 1st driver car interaction and interface (DCII 2008)*, pages 35–60, 2009.
- [24] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017.
- [25] N. Jaques, C. Conati, J. M. Harley, and R. Azevedo. Predicting affect from gaze data during interaction with an intelligent tutoring system. In *International Conference on Intelligent Tutoring Systems*, pages 29–38. Springer, 2014.
- [26] S. Kai, L. Paquette, R. S. Baker, N. Bosch, S. D’Mello, J. Ocumpaugh, V. Shute, and M. Ventura. A comparison of video-based and interaction-based affect detectors in physics playground. *International Educational Data Mining Society*, 2015.
- [27] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR*, 2018.
- [28] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.
- [29] V. Kostyuk, M. V. Almeda, and R. S. Baker. Correlating affect and behavior in reasoning mind with state test achievement. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, pages 26–30. ACM, 2018.
- [30] P. J. Lang, M. M. Bradley, and B. N. Cuthbert. International affective picture system (IAPS): Affective ratings of pictures and instruction manual. Technical Report A-8, The Center for Research in Psychophysiology, University of Florida, Gainesville, FL, 2008.
- [31] Y. Li, S. Liu, J. Yang, and M.-H. Yang. Generative face completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3911–3919, 2017.
- [32] H. Liao, G. Funka-Lea, Y. Zheng, J. Luo, and K. S. Zhou. Face completion with semantic knowledge and collaborative adversarial learning. In *Asian Conference on Computer Vision*, pages 382–397. Springer, 2018.
- [33] D. J. Litman and K. Forbes-Riley. Recognizing student

- emotions and attitudes on the basis of utterances in spoken tutoring dialogues with both human and computer tutors. *Speech communication*, 48(5):559–590, 2006.
- [34] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.
- [35] H. Liu, B. Jiang, Y. Xiao, and C. Yang. Coherent semantic attention for image inpainting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4170–4179, 2019.
- [36] K. Lochner and M. Eid. *Successful emotions: how emotions drive cognitive performance*. Springer, 2016.
- [37] D. Malesevic, C. Mayer, S. Gu, and R. Timofte. Photo-realistic and robust inpainting of faces using refinement gans. In *Inpainting and Denoising Challenges*, pages 129–144. Springer, 2019.
- [38] B. McDaniel, S. D’Mello, B. King, P. Chipman, K. Tapp, and A. Graesser. Facial features for affective state detection in learning environments. *Proceedings of the 29th Annual Meeting of the Cognitive Science Society*, pages 467–472, 2007.
- [39] D. McDuff, A. Mahmoud, M. Mavadati, M. Amr, J. Turcot, and R. Kaliouby. Affdex sdk: a cross-platform real-time multi-face expression recognition toolkit. In *Proceedings of the 2016 CHI conference extended abstracts on human factors in computing systems*, pages 3723–3726, 2016.
- [40] A. Mehrabian and J. A. Russell. *An approach to environmental psychology*. MIT Press, 1974.
- [41] M. Miserandino. Children who do well in school: Individual differences in perceived competence and autonomy in above-average children. *Journal of educational psychology*, 88(2):203, 1996.
- [42] R. Navarathna, P. Lucey, P. Carr, E. Carter, S. Sridharan, and I. Matthews. Predicting movie ratings from audience behaviors. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1058–1065. IEEE, 2014.
- [43] B. T. Nguyen, M. H. Trinh, T. V. Phan, and H. D. Nguyen. An efficient real-time emotion detection using camera and facial landmarks. In *2017 Seventh International Conference on Information Science and Technology (ICIST)*, pages 251–255. IEEE, 2017.
- [44] Z. A. Pardos, R. S. J. D. Baker, M. O. C. Z. San Pedro, S. M. Gowda, and S. M. Gowda. Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. In *Proc. LAK*, pages 117–124. ACM, 2013.
- [45] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.
- [46] P. Pham and J. Wang. Predicting learners’ emotions in mobile mooc learning via a multimodal intelligent tutor. In *International Conference on Intelligent Tutoring Systems*, pages 150–159. Springer, 2018.
- [47] D. Reid and N. Ostashevski. iPads in the classroom—new technologies, old issues: Are they worth the effort? In *EdMedia+ Innovate Learning*, pages 1689–1694. Association for the Advancement of Computing in Education (AACE), 2011.
- [48] J. A. Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.
- [49] S. Salmeron-Majadas, R. S. Baker, O. C. Santos, and J. G. Boticario. A machine learning approach to leverage individual keyboard and mouse interaction behavior from multiple users in real-world learning scenarios. *IEEE Access*, 6:39154–39179, 2018.
- [50] G. K. Sarpate and S. K. Guru. Image inpainting on satellite image using texture synthesis & region filling algorithm. In *2014 International Conference on Advances in Communication and Computing Technologies (ICACACT 2014)*, pages 1–5. IEEE, 2014.
- [51] K. R. Scherer. What are emotions? and how can they be measured? *Social science information*, 44(4):695–729, 2005.
- [52] A. Singh, C. Chandewar, and P. Pattarkine. Driver drowsiness alert system with effective feature extraction. *International Journal for Research in Emerging Science and Technology*, 5(4):26–31, 2018.
- [53] R. Wampfler, S. Klingler, B. Solenthaler, V. Schinazi, and M. Gross. Affective state prediction in a mobile setting using wearable biometric sensors and stylus. In *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, pages 198–207, 2019.
- [54] C. Wang, X. Sun, F. Wu, and H. Xiong. Image compression with structure-aware inpainting. In *2006 IEEE International Symposium on Circuits and Systems*, pages 4–pp. IEEE, 2006.
- [55] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5485–5493, 2017.
- [56] J. Zaletelj and A. Košir. Predicting students’ attention in the classroom from kinect facial and body features. *EURASIP journal on image and video processing*, 2017(1):80, 2017.
- [57] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE transactions on pattern analysis and machine intelligence*, 31(1):39–58, 2008.
- [58] Y. Zhuang, Y. Wang, T. K. Shih, and N. C. Tang. Patch-guided facial image inpainting by shape propagation. *Journal of Zhejiang University-SCIENCE A*, 10(2):232–238, 2009.

Variational Item Response Theory: Fast, Accurate, and Expressive

Mike Wu¹, Richard L. Davis², Benjamin W. Domingue², Chris Piech¹, Noah Goodman^{1,3}

Department of Computer Science¹, Education², and Psychology³
Stanford University

{wumike, rldavis, bdomingu, cpiech, ngoodman}@stanford.edu

ABSTRACT

Item Response Theory (IRT) is a ubiquitous model for understanding humans based on their responses to questions, used in fields as diverse as education, medicine and psychology. Large modern datasets offer opportunities to capture more nuances in human behavior, potentially improving test scoring and better informing public policy. Yet larger datasets pose a difficult speed / accuracy challenge to contemporary algorithms for fitting IRT models. We introduce a variational Bayesian inference algorithm for IRT, and show that it is fast and scaleable without sacrificing accuracy. Using this inference approach we then extend classic IRT with expressive Bayesian models of responses. Applying this method to five large-scale item response datasets from cognitive science and education yields higher log likelihoods and improvements in imputing missing data. The algorithm implementation is open-source, and easily usable.

1. INTRODUCTION

The task of estimating human ability from stochastic responses to a series of questions has been studied since the 1950s in thousands of papers spanning several fields. The standard statistical model for this problem, Item Response Theory (IRT), is used every day around the world, in many critical contexts including college admissions tests, school-system assessment, survey analysis, popular questionnaires, and medical diagnosis.

As datasets become larger, new challenges and opportunities for improving IRT models present themselves. On the one hand, massive datasets offer the opportunity to better understand human behavior, fitting more expressive models. On the other hand, the algorithms that work for fitting small datasets often become intractable for larger data sizes. Indeed, despite a large body of literature, contemporary IRT methods fall short – it remains surprisingly difficult to estimate human ability from stochastic responses. One crucial bottleneck is that the most accurate, state-of-the-art Bayesian inference algorithms are prohibitively slow, while

faster algorithms (such as the popular maximum marginal likelihood estimators) are less accurate and poorly capture uncertainty. This leaves practitioners with a choice: either have nuanced Bayesian models with appropriate inference or have timely computation.

In the field of artificial intelligence, a revolution in deep generative models via *variational inference* [25, 37] has demonstrated an impressive ability to perform fast inference for complex Bayesian models. In this paper, we present a novel application of variational inference to IRT, validate the resulting algorithms with synthetic datasets, and apply them to real world datasets. We then show that this inference approach allows us to extend classic IRT response models with deep neural network components. We find that these more flexible models better fit the large real world datasets. Specifically, our contributions are as follows:

1. **Variational inference for IRT:** We derive a new optimization objective — the Variational Item response theory Lower Bound, or VIBO — to perform inference in IRT models. By learning a mapping from responses to posterior distributions over ability and items, VIBO is “amortized” to solve inference queries efficiently.
2. **Faster inference:** We find VIBO to be much faster than previous Bayesian techniques and usable on much larger datasets without loss in accuracy.
3. **More expressive:** Our inference approach is naturally compatible with deep generative models and, as such, we enable the novel extension of Bayesian IRT models to use neural-network-based representations for inputs, predictions, and student ability. We develop the first deep generative IRT models.
4. **Simple code:** Using our VIBO python package¹ is only a few lines of code that is easy to extend.
5. **Real world application:** We demonstrate the impact of faster inference and expressive models by applying our algorithms to datasets including: PISA, DuoLingo and Gradescope. We achieve up to 200 times speedup and show improved accuracy at imputing hidden responses. At scale, these improvements in efficiency save hundreds of hours of computation.

Mike Wu, Richard Davis, Benjamin Domingue, Chris Piech and Noah Goodman "Variational Item Response Theory: Fast, Accurate, and Expressive" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 257 - 268

¹<http://github.com/mhw32/variational-item-response-theory-public>

As a roadmap, in Sec. 2 we describe the item response theory challenge. In Sec. 3 we present a main algorithm. Finally, in Sec. 4 and 5 we show its impact on speed and accuracy.

2. BACKGROUND

We briefly review several variations of item response theory and the fundamental principles of approximate Bayesian inference, focusing on modern variational inference.

2.1 Item Response Theory Review

Imagine answering a series of multiple choice questions. For example, consider a personality survey, a homework assignment, or a school entrance examination. Selecting a response to each question is an interaction between your “ability” (knowledge or features) and the characteristics of the question, such as its difficulty. The goal in examination analysis is to gauge this unknown ability of each student and the unknown item characteristics based only on responses. Early procedures [11] defaulted to very simple methods, such as counting the number of correct responses, which ignore differences in question quality. In reality, we understand that not all questions are created equal: some may be hard to understand while others may test more difficult concepts. To capture these nuances, Item Response Theory (IRT) was developed as a mathematical framework to reason jointly about people’s ability and the items themselves.

The IRT model plays an impactful role in many large institutions. It is the preferred method for estimating ability in several state assessments in the United States, for international assessments gauging educational competency across countries [18], and for the National Assessment of Educational Programs (NAEP), a large-scale measurement of literacy and reading comprehension in the US [35]. Beyond education, IRT is a method widely used in cognitive science and psychology, for instance with regards to studies of language acquisition and development [19, 28, 14, 7].

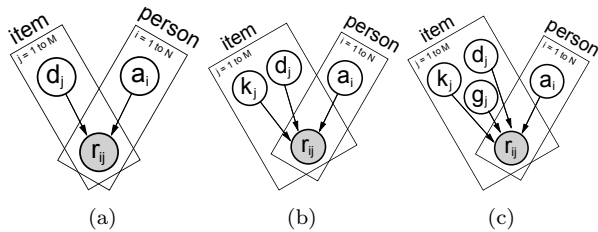


Figure 1: Graphical models for the (a) 1PL, (b) 2PL, and (c) 3PL Item Response Theories. Observed variables are shaded. Arrows represent dependency between random variables and each rectangle represents a plate (i.e. repeated observations).

IRT has many forms; we review the most standard (Fig. 1). The simplest class of IRT summarizes the ability of a person with a single parameter. This class contains three versions: 1PL, 2PL, and 3PL IRT, each of which differ by the number of free variables used to characterize an item. The 1PL IRT model, also called the Rasch model [34], is given in Eq. 1,

$$p(r_{i,j} = 1 | a_i, d_j) = \frac{1}{1 + e^{-(a_i - d_j)}} \quad (1)$$

where $r_{i,j}$ is the response by the i -th person to the j -th item. There are N people and M items in total. Each

item in the 1PL model is characterized by a single number representing difficulty, d_j . As the 1PL model is equivalent to a logistic function, a higher difficulty requires a higher ability in order to respond correctly. Next, the 2PL IRT model² adds a *discrimination* parameter, k_j for each item that controls the slope (or scale) of the logistic curve. We can expect items with higher discrimination to more quickly separate people of low and high ability. The 3PL IRT model further adds a *pseudo-guessing* parameter, g_j for each item that sets the asymptotic minimum of the logistic curve. We can interpret pseudo-guessing as the probability of success if the respondent were to make a reasonable guess on an item. The 2PL and 3PL IRT models are:

$$p(r_{i,j} | a_i, \mathbf{d}_j) = \frac{1}{1 + e^{-k_j a_i - d_j}} \text{ or } g_j + \frac{1 - g_j}{1 + e^{-k_j a_i - d_j}} \quad (2)$$

where $\mathbf{d}_j = \{k_j, d_j\}$ for 2PL and $\mathbf{d}_j = \{g_j, k_j, d_j\}$ for 3PL. See Fig. 1 for graphical models of each of these IRT models.

A single ability dimension is sometimes insufficient to capture the relevant variation in human responses. For instance, if we are measuring a person’s understanding on elementary arithmetic, then a single dimension may suffice in capturing the majority of the variance. However, if we are instead measuring a person’s general mathematics ability, a single real number no longer seems sufficient. Even if we bound the domain to middle school mathematics, there are several factors that contribute to “mathematical understanding” (e.g. proficiency in algebra versus geometry). Summarizing a person with a single number in this setting would result in a fairly loose approximation. For cases where multiple facets of ability contribute to performance, we consider *multidimensional* item response theory [1, 36, 29]. We focus on 2PL multidimensional IRT (MIRT):

$$p(r_{i,j} = 1 | \mathbf{a}_i, \mathbf{k}_j, d_j) = \frac{1}{1 + e^{-\mathbf{a}_i^T \mathbf{k}_j - d_j}} \quad (3)$$

where we use bolded notation $\mathbf{a}_i = (a_i^{(1)}, a_i^{(2)}, \dots, a_i^{(K)})$ to represent a K dimensional vector. Notice that the item discrimination becomes a vector of equal size to ability.

In practice, given a (possibly incomplete) $N \times M$ matrix of observed responses, we want to *infer* the ability of all N people and the characteristics of all M items. Next, we provide a brief overview of inference in IRT.

2.2 Inference in Item Response Theory

Inference is the task of estimating unknown variables, such as ability, given observations, such as student responses. We compare and contrast three popular methods used to perform inference for IRT in research and industry. Inference algorithms are critical for item response theory as slow or inaccurate algorithms prevent the use of appropriate models.

Maximum Likelihood Estimation A straightforward approach is to pick the most likely ability and item features

²We default to 2PL as the pseudo-guessing parameter introduces several invariances in the model. This requires far more data to infer ability accurately, as measured by our own synthetic experiments. For practitioners, we warn against using 3PL for small to medium datasets.

given the observed responses. To do so we optimize:

$$\mathcal{L}_{\text{MLE}} = \max_{\{\mathbf{a}_i\}_{i=1}^N, \{\mathbf{d}_j\}_{j=1}^M} \sum_{i=1}^N \sum_{j=1}^M \log p(r_{ij} | \mathbf{a}_i, \mathbf{d}_j) \quad (4)$$

with stochastic gradient descent (SGD). The symbol \mathbf{d}_j represents all item features e.g. $\mathbf{d}_j = \{d_j, \mathbf{k}_j\}$ for 2PL. Eq. 4 is often called the Joint Maximum Likelihood Estimator [3, 12], abbreviated MLE. MLE poses inference as a supervised regression problem in which we choose the most likely unknown variables to match known dependent variables. While MLE is simple to understand and implement, it lacks any measure of uncertainty; this can have important consequences especially when responses are missing.

Expectation Maximization Several papers have pointed out that when using MLE, the number of unknown parameters increases with the number of people [5, 17]. In particular, [17] shows that in practical settings with a finite number of items, standard convergence theorems do not hold for MLE as the number of people grows. To remedy this, the authors instead treat ability as a nuisance parameter and marginalized it out [5, 6]. Brock et. al. introduces an Expectation-Maximization (EM) [10] algorithm to iterate between (1) updating beliefs about item characteristics and (2) using the updated beliefs to define a marginal distribution (without ability) $p(r_{ij} | \mathbf{d}_j)$ by numerical integration of \mathbf{a}_i . Appropriately, this algorithm is referred to as Maximum Marginal Likelihood Estimation, which we abbreviate as EM. Eq. 6 shows the E and M steps for EM.

$$\text{E step: } p(r_{ij} | \mathbf{d}_j^{(t)}) = \int_{\mathbf{a}_i} p(r_{ij} | \mathbf{a}_i, \mathbf{d}_j^{(t)}) p(\mathbf{a}_i) d\mathbf{a}_i \quad (5)$$

$$\text{M step: } \mathbf{d}_j^{(t+1)} = \arg \max_{\mathbf{d}_j} \sum_{i=1}^N \log p(r_{ij} | \mathbf{d}_j^{(t)}) \quad (6)$$

where (t) represents the iteration count. We often choose $p(\mathbf{a}_i)$ to be a simple prior distribution like standard Normal. In general, the integral in the E-step is intractable: EM uses a Gaussian-Hermite quadrature to discretely approximate $p(r_{ij} | \mathbf{d}_j^{(t)})$. See [20] for a closed form expression for $\mathbf{d}_j^{(t+1)}$ in the M step. This method finds the maximum a posteriori (MAP) estimate for item characteristics. EM does not infer ability as it is “ignored” in the model: the common workaround is to use EM to infer item characteristics, then fit ability using a second auxiliary model. In practice, EM has grown to be ubiquitous in industry as it is incredibly fast for small to medium sized datasets. However, we expect that EM may scale poorly to large datasets and higher dimensions as numerical integration requires far more points to properly measure a high dimensional volume.

Hamiltonian Monte Carlo The two inference methods above give only point estimates for ability and item characteristics. In contrast Bayesian approaches seek to capture the full posterior over ability and item characteristics given observed responses, $p(\mathbf{a}_i, \mathbf{d}_{1:M} | \mathbf{r}_{i,1:M})$ where $\mathbf{r}_{i,1:M} = (r_{i,1}, \dots, r_{i,M})$. Doing so provides estimates of uncertainty and characterizes features of the joint distribution that cannot be represented by point estimates, such as multimodality and parameter correlation. In practice, this can be very useful for a more robust understanding of student ability.

The common technique for Bayesian estimation in IRT uses Markov Chain Monte Carlo (MCMC) [21, 15] to draw samples from the posterior by constructing a Markov chain carefully designed such that $p(\mathbf{a}_i, \mathbf{d}_{1:M} | \mathbf{r}_{i,1:M})$ is the equilibrium distribution. By running the chain longer, we can closely match the distribution of drawn samples to the true posterior. Hamiltonian Monte Carlo (HMC) [33, 32, 23] is an efficient version of MCMC for continuous state spaces. We recommend [23] for a good review of HMC.

The strength of this approach is that the samples generated capture the true posterior (if the algorithm is run long enough). But the computational costs for MCMC can be very high, and the cost scales at least linearly with the number of latent parameters — which for IRT is proportional to data size. With new datasets of millions of observations, such limitations can be debilitating. Fortunately, there exist a second class of approximate Bayesian techniques that have gained significant traction in the machine learning community. We provide a careful review of *variational inference*.

2.3 Variational Methods Review

Variational inference (VI) first appeared from the statistical physics community and was later generalized for many probabilistic models by Jordan et. al. [24]. In recent years, VI has been popularized in machine learning where it is used to do inference in large graphical models describing images and natural language. The main intuition of variational inference is to treat inference as an optimization problem: starting with a family of distributions, the goal is to pick the one that best approximates the true posterior, by minimizing an estimate of the mismatch between true and approximate distributions. We will first describe VI in the general context of a latent variable model, and then apply VI to IRT.

Let $\mathbf{x} \in \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z}$ represent observed and latent variables, respectively. (In the context of IRT, \mathbf{x} represents the responses from a single student and \mathbf{z} represents ability and item characteristics.) In VI [24, 42, 4], we introduce a family of tractable distributions over \mathbf{z} such that we can easily sample from and score. We wish to find the member $q_{\psi^*(\mathbf{x})} \in \mathcal{Q}$ that minimizes the Kullback-Leibler (KL) divergence between itself and the true posterior:

$$q_{\psi^*(\mathbf{x})}(\mathbf{z}) = \arg \min_{q_{\psi(\mathbf{x})}} D_{\text{KL}}(q_{\psi(\mathbf{x})}(\mathbf{z}) || p(\mathbf{z} | \mathbf{x})) \quad (7)$$

where $\psi(\mathbf{x})$ are parameters that define each distribution. For example, $\psi(\mathbf{x})$ would be the mean and scale for a Gaussian distribution. Since the “best” approximate posterior $q_{\psi^*(\mathbf{x})}$ depends on the observed variables, its parameters have \mathbf{x} as a dependent variable. To be clear, there is one approximate posterior for every possible value of the observed variables.

Frequently, we need to do inference for many different values of \mathbf{x} . For example, student A and student B may have picked different answers to the same question. Since their responses differ, we would need to do inference twice. Let $p_{\mathcal{D}}(\mathbf{x})$ be an empirical distribution over the observed variables, which is equivalent to the marginal $p(\mathbf{x})$ if the generative model is correctly specified. Then, the average quality of the variational approximations is measured by

$$\mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})} \left[\max_{\psi(\mathbf{x})} \mathbb{E}_{q_{\psi(\mathbf{x})}(\mathbf{z})} \left[\log \frac{p(\mathbf{x}, \mathbf{z})}{q_{\psi(\mathbf{x})}(\mathbf{z})} \right] \right] \quad (8)$$

In practice, $p_{\mathcal{D}}(\mathbf{x})$ is unknown but we assume access to a dataset \mathcal{D} of examples i.i.d. sampled from $p_{\mathcal{D}}(\mathbf{x})$; this is sufficient to evaluate Eq. 8.

Amortization As in Eq. 8, we must learn an approximate posterior for each $\mathbf{x} \in \mathcal{D}$. For a large dataset \mathcal{D} , this can quickly grow to be unwieldy. One such solution to this scalability problem is *amortization* [16], which reframes the per-observation optimization problem as a supervised regression task. Consider learning a single deterministic mapping $f_{\phi} : \mathcal{X} \rightarrow \mathcal{Q}$ to predict $\psi^*(\mathbf{x})$ or equivalently $q_{\psi^*(\mathbf{x})} \in \mathcal{Q}$ as a function of the observation \mathbf{x} . Often, we choose f_{ϕ} to be a conditional distribution, denoted by $q_{\phi}(\mathbf{z}|\mathbf{x}) = f_{\phi}(\mathbf{x})(\mathbf{z})$.

The benefit of amortization is a large reduction in computational cost: the number of parameters is vastly smaller than learning a per-observation posterior. Additionally, if we manage to learn a good regressor, then the amortized approximate posterior $q_{\phi}(\mathbf{z}|\mathbf{x})$ could generalize to new observations $\mathbf{x} \notin \mathcal{D}$ unseen in training. This strength has made amortized VI popular with modern latent variable models, such as the Variational Autoencoder [25].

Instead of Eq. 8, we now optimize:

$$\max_{\phi} \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})} \left[\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[\log \frac{p(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right] \right] \quad (9)$$

The drawback of this approach is that it introduces an *amortization gap*: since we are technically using a less flexible family of approximate distributions, the quality of approximate posteriors can be inferior.

Model Learning So far we have assumed a fixed generative model $p(\mathbf{x}, \mathbf{z})$. However, often we can only specify a family of possible models $p_{\theta}(\mathbf{x}|\mathbf{z})$ parameterized by θ . The symmetric challenge (to approximate inference) is to choose θ whose model best explains the evidence. Naturally, we do so by maximizing the log marginal likelihood of the data

$$\log p_{\theta}(\mathbf{x}) = \log \int_{\mathbf{z}} p_{\theta}(\mathbf{x}, \mathbf{z}) d\mathbf{z} \quad (10)$$

Using Eq. 9, we derive the Evidence Lower Bound (ELBO) [25, 37] with $q_{\phi}(\mathbf{z}|\mathbf{x})$ as our inference model

$$\log p_{\theta}(\mathbf{x}) \geq \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right] \triangleq \text{ELBO} \quad (11)$$

We can jointly optimize ϕ and θ to maximize the ELBO. We have the option to parameterize $p_{\theta}(\mathbf{x}|\mathbf{z})$ and $q_{\phi}(\mathbf{z}|\mathbf{x})$ with deep neural networks, as is common with the VAE [25], yielding an extremely flexible space of distributions.

Stochastic Gradient Estimation The gradients of the ELBO (Eq. 11) with respect to ϕ and θ are:

$$\nabla_{\theta} \text{ELBO} = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\nabla_{\theta} \log p_{\theta}(\mathbf{x}, \mathbf{z})] \quad (12)$$

$$\nabla_{\phi} \text{ELBO} = \nabla_{\phi} \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}, \mathbf{z})] \quad (13)$$

Eq. 12 can be estimated using Monte Carlo samples. However, as it stands, Eq. 13 is difficult to estimate as we cannot distribute the gradient inside the inner expectation. For certain families \mathcal{Q} , we can use a reparameterization trick.

Reparameterization Estimators Reparameterization is the technique of removing sampling from the gradient

computation graph [25, 37]. In particular, if we can reduce sampling $\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})$ to sampling from a parameter-free distribution $\epsilon \sim p(\epsilon)$ plus a deterministic function application, $\mathbf{z} = g_{\phi}(\epsilon)$, then we may rewrite Eq. 13 as:

$$\nabla_{\phi} \text{ELBO} = \mathbb{E}_{p(\epsilon)} [\nabla_{\mathbf{z}} \log \frac{p_{\theta}(\mathbf{x}, g_{\phi}(\epsilon))}{q_{\phi}(g_{\phi}(\epsilon)|\mathbf{x})} \nabla_{\phi} g_{\phi}(\epsilon)] \quad (14)$$

which now can be estimated efficiently by Monte Carlo (the gradient is inside the expectation). A benefit of reparameterization over alternative estimators (e.g. score estimator [30] or REINFORCE [44]) is lower variance while remaining unbiased. A common example is if $q_{\phi}(\mathbf{z}|\mathbf{x})$ is Gaussian $\mathcal{N}(\mu, \sigma^2)$ and we choose $p(\epsilon)$ to be $\mathcal{N}(0, 1)$, then $g(\epsilon) = \epsilon * \sigma + \mu$.

3. THE VIBO ALGORITHM

Having rehearsed the major principles of VI, we will adapt them to IRT. In our review, we presented the ELBO that serves as the primary loss function to train an inference model. Given the nuances of IRT, we can derive a new loss function specialized for ability and item characteristics. We call the resulting algorithm VIBO since it is a **V**ariational approach for **I**tem response theory based on a novel lower **B**ound. While the remainder of the section presents the technical details, we ask the reader to keep the high-level purpose in mind: VIBO is an objective function that if we maximize, we have a method to predict student ability from his or her responses. As an optimization problem, VIBO is much cheaper computationally than MCMC.

To show that doing so is justifiable, we prove that VIBO well-defined. That is, we must show that VIBO lower bounds the marginal likelihood over a student's responses.

THEOREM 3.1. *Let \mathbf{a}_i be the ability for person $i \in [1, N]$ and \mathbf{d}_j be the characteristics for item $j \in [1, M]$. We use the shorthand notation $\mathbf{d}_{1:M} = (\mathbf{d}_1, \dots, \mathbf{d}_M)$. Let $r_{i,j}$ be the binary response for item j by person i . We write $\mathbf{r}_{i,1:M} = (r_{i,1}, \dots, r_{i,M})$. If we define the VIBO objective as:*

$$\text{VIBO} \triangleq \mathcal{L}_{\text{recon}} + \mathbb{E}_{q_{\phi}(\mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} [D_{\text{ability}}] + D_{\text{item}}$$

where

$$\mathcal{L}_{\text{recon}} = \mathbb{E}_{q_{\phi}(\mathbf{a}_i, \mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} [\log p_{\theta}(\mathbf{r}_{i,1:M}|\mathbf{a}_i, \mathbf{d}_{1:M})]$$

$$D_{\text{ability}} = D_{\text{KL}}(q_{\phi}(\mathbf{a}_i|\mathbf{d}_{1:M}, \mathbf{r}_{i,1:M}) || p(\mathbf{a}_i))$$

$$D_{\text{item}} = D_{\text{KL}}(q_{\phi}(\mathbf{d}_{1:M}|\mathbf{r}_{i,1:M}) || p(\mathbf{d}_{1:M}))$$

and assume the joint posterior factors as $q_{\phi}(\mathbf{a}_i, \mathbf{d}_{1:M}|\mathbf{r}_{i,1:M}) = q_{\phi}(\mathbf{a}_i|\mathbf{d}_{1:M}, \mathbf{r}_{i,1:M}) q_{\phi}(\mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})$, then $\log p(\mathbf{r}_{i,1:M}) \geq \text{VIBO}$. In other words, VIBO is a lower bound on the log marginal probability of person i 's responses.

PROOF. Expand marginal and apply Jensen's inequality:

$$\begin{aligned} \log p_{\theta}(\mathbf{r}_{i,1:M}) &\geq \mathbb{E}_{q_{\phi}(\mathbf{a}_i, \mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} \left[\log \frac{p_{\theta}(\mathbf{r}_{i,1:M}, \mathbf{a}_i, \mathbf{d}_{1:M})}{q_{\phi}(\mathbf{a}_i, \mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} \right] \\ &= \mathbb{E}_{q_{\phi}(\mathbf{a}_i, \mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} [\log p_{\theta}(\mathbf{r}_{i,1:M}|\mathbf{a}_i, \mathbf{d}_{1:M})] \\ &\quad + \mathbb{E}_{q_{\phi}(\mathbf{a}_i, \mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} \left[\log \frac{p(\mathbf{a}_i)}{q_{\phi}(\mathbf{a}_i|\mathbf{d}_{1:M}, \mathbf{r}_{i,1:M})} \right] \\ &\quad + \mathbb{E}_{q_{\phi}(\mathbf{a}_i, \mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} \left[\log \frac{p(\mathbf{d}_{1:M})}{q_{\phi}(\mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} \right] \\ &= \mathcal{L}_{\text{recon}} + \mathcal{L}_A + \mathcal{L}_B \end{aligned}$$

Rearranging the latter two terms, we find that:

$$\begin{aligned}\mathcal{L}_A &= \mathbb{E}_{q_\phi(\mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} [D_{\text{KL}}(q_\phi(\mathbf{a}_i|\mathbf{d}_{1:M}, \mathbf{r}_{i,1:M})||p(\mathbf{a}_i))] \\ \mathcal{L}_B &= \mathbb{E}_{q_\phi(\mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} \left[\log \frac{p(\mathbf{d}_{1:M})}{q_\phi(\mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} \right] \\ &= D_{\text{KL}}(q_\phi(\mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})||p(\mathbf{d}_{1:M}))\end{aligned}$$

Since $\text{VIBO} = \mathcal{L}_{\text{recon}} + \mathcal{L}_A + \mathcal{L}_B$, and KL terms are non-negative, we have shown that VIBO bounds $\log p_\theta(\mathbf{r}_{i,1:M})$. \square

Thm. 3.1 leaves several choices up to us, and we opt for the simplest ones. For instance, the prior distributions are chosen to be independent standard Normal distributions: $p(\mathbf{a}_i) = \prod_{k=1}^K p(a_{i,k})$ and $p(\mathbf{d}_{1:M}) = \prod_{j=1}^M p(\mathbf{d}_j)$ where $p(a_{i,k})$ and $p(\mathbf{d}_j)$ are $\mathcal{N}(0, 1)$. Further, we found it sufficient to assume $q_\phi(\mathbf{d}_{1:M}|\mathbf{r}_{i,1:M}) = q_\phi(\mathbf{d}_{1:M}) = \prod_{j=1}^M q_\phi(\mathbf{d}_j)$ although nothing prevents the general case. Initially, we assume the generative model, $p_\theta(\mathbf{r}_{i,1:M}|\mathbf{a}_i, \mathbf{d}_{1:M})$, to be an IRT model (thus θ is empty); later we explore generalizations.

Algorithm 1: VIBO Forward Pass

Assume we are given observed responses for person i , $\mathbf{r}_{i,1:M}$;
 Compute $\mu_{\mathbf{a}}, \sigma_{\mathbf{a}}^2 = q_\phi(\mathbf{d}_{1:M})$;
 Sample $\mathbf{d}_{1:M} \sim \mathcal{N}(\mu_{\mathbf{a}}, \sigma_{\mathbf{a}}^2)$;
 Compute $\mu_{\mathbf{a}}, \sigma_{\mathbf{a}}^2 = q_\phi(\mathbf{a}_i|\mathbf{d}_{1:M}, \mathbf{r}_{i,1:M})$;
 Sample $\mathbf{a}_i \sim \mathcal{N}(\mu_{\mathbf{a}}, \sigma_{\mathbf{a}}^2)$;
 Compute $\mathcal{L}_{\text{recon}} = \log p_\theta(\mathbf{r}_{i,1:M}|\mathbf{a}_i, \mathbf{d}_{1:M})$;
 Compute $D_{\text{ability}} = D_{\text{KL}}(\mathcal{N}(\mu_{\mathbf{a}}, \sigma_{\mathbf{a}}^2)||\mathcal{N}(0, 1))$;
 Compute $D_{\text{item}} = D_{\text{KL}}(\mathcal{N}(\mu_{\mathbf{d}}, \sigma_{\mathbf{d}}^2)||\mathcal{N}(0, 1))$;
 Compute $\text{VIBO} = \mathcal{L}_{\text{recon}} + D_{\text{ability}} + D_{\text{item}}$

The posterior $q_\phi(\mathbf{a}_i|\mathbf{d}_{1:M}, \mathbf{r}_{i,1:M})$ needs to be robust to missing data as often not every person answers every question. To achieve this, we explore the following family:

$$q_\phi(\mathbf{a}_i|\mathbf{d}_{1:M}, \mathbf{r}_{i,1:M}) = \prod_{j=1}^M q_\phi(\mathbf{a}_i|\mathbf{d}_j, \mathbf{r}_{i,j}) \quad (15)$$

If we assume each component $q_\phi(\mathbf{a}_i|\mathbf{d}_j, \mathbf{r}_{i,j})$ is Gaussian, then $q_\phi(\mathbf{a}_i|\mathbf{d}_{1:M}, \mathbf{r}_{i,1:M})$ is Gaussian as well, being a Product-Of-Experts [22, 45]. If item j is missing, we replace its term in the product with the prior, $p(\mathbf{a}_i)$ representing no added information. We found this design to outperform averaging over non-missing entries: $\frac{1}{M} \sum_{j=1}^M q_\phi(\mathbf{a}_i|\mathbf{d}_j, \mathbf{r}_{i,j})$.

As VIBO is a close cousin of the ELBO, we can estimate its gradients with respect to θ and ϕ similarly:

$$\begin{aligned}\nabla_\theta \text{VIBO} &= \nabla_\theta \mathcal{L}_{\text{recon}} \\ &= \mathbb{E}_{q_\phi(\mathbf{a}_i, \mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} [\nabla_\theta \log p_\theta(\mathbf{r}_{i,1:M}|\mathbf{a}_i, \mathbf{d}_{1:M})] \\ \nabla_\phi \text{VIBO} &= \nabla_\phi \mathbb{E}_{q_\phi(\mathbf{a}_i, \mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} [D_{\text{ability}}] + \nabla_\phi D_{\text{item}} \\ &= \nabla_\phi \mathbb{E}_{q_\phi(\mathbf{a}_i, \mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} \left[\frac{p(\mathbf{a}_i)p(\mathbf{d}_{1:M})}{q_\phi(\mathbf{a}_i, \mathbf{d}_{1:M}|\mathbf{r}_{i,1:M})} \right]\end{aligned}$$

As in Eq. 14, we may wish to move the gradient inside the KL divergences by reparameterization to reduce variance. To allow easy reparameterization, we define all variational distributions $q_\phi(\cdot|\cdot)$ as Normal distributions with diagonal covariance. In practice, we find that estimating $\nabla_\theta \text{VIBO}$ and $\nabla_\phi \text{VIBO}$ with a single sample is sufficient. With this setup,

VIBO can be optimized using stochastic gradient descent to learn an amortized inference model that maximizes the marginal probability of observed data. We summarize the required computation to calculate VIBO in Alg. 1.

4. DATASETS

We explore one synthetic dataset, to build intuition and confirm parameter recovery, and five large scale applications of IRT to real world data, summarized in Table 1.

Table 1: Dataset Statistics

	# PERSONS	# ITEMS	MISSING DATA?
CRITLANGACQ	669498	95	N
WORDBANK	5520	797	N
DUOLINGO	2587	2125	Y
GRADESCOPE	1254	98	Y
PISA	519334	183	Y

Synthetic IRT To sanity check that VIBO performs as well as other inference techniques, we synthetically generate a dataset of responses using a 2PL IRT model: sample $\mathbf{a}_i \sim p(\mathbf{a}_i)$, $\mathbf{d}_j \sim p(\mathbf{d}_j)$. Given ability and item characteristics, IRT-2PL determines a Bernoulli distribution over responses to item j by person i . We sample once from this Bernoulli distribution to “generate” an observation. In this setting, we know the ground truth ability and item characteristics. We vary N and M to explore parameter recovery.

Second Language Acquisition This dataset contains native and non-native English speakers answering questions to a grammar quiz³, which upon completion would return a prediction of the user’s native language. Using social media, over half a million users of varying ages and demographics completed the quiz. Quiz questions often contain both visual and linguistic components. For instance, a quiz question could ask the user to “choose the image where the dog is chased by the cat” and present two images of animals where only one of image agrees with the caption. Every response is thus binary, marked as correct or incorrect. In total, there are 669,498 people with 95 items and no missing data. The creators of this dataset use it to study the presence or absence of a “critical period” for second language acquisition [19]. We will refer to this dataset as CRITLANGACQ.

WordBank: Vocabulary Development The MacArthur-Bates Communicative Development Inventories (CDIs) are a widely used metric for early language acquisition in children, testing concepts in vocabulary comprehension, production, gestures, and grammar. The WordBank [14] database archives many independently collected CDI datasets across languages and research laboratories⁴. The database consists of a matrix of people against vocabulary words where the (i, j) entry is 1 if a parent reports that child i has knowledge of word j and 0 otherwise. Some entries are missing due to slight variations in surveys and incomplete responses. In total, there are 5,520 children responding to 797 items.

³The quiz can be found at www.gameswithwords.org. The data is publically available at osf.io/pyb8s.

⁴github.com/langcog/wordbankr

DuoLingo: App-Based Language Learning We examine the 2018 DuoLingo Shared Task on Second Language Acquisition Modeling⁵ [38]. This dataset contains anonymized user data from the popular education application, DuoLingo. In the application, users must choose the correct vocabulary word among a list of distractor words. We focus on the subset of native English speakers learning Spanish and only consider lesson sessions. Each user has a timeseries of responses to a list of vocabulary words, each of which is shown several times. We repurpose this dataset for IRT: the goal being to infer the user’s language proficiency from his or her errors. As such, we average over all times a user has seen each vocabulary item. For example, if the user was presented “habla” 10 times and correctly identified the word 5 times, he or she would be given a response score of 0.5. We then round to 0 or 1. We revisit a continuous version Sec. 7. After processing, we have 2587 users and 2125 vocabulary words with missing data as users frequently drop out. We ignore user and syntax features.

Gradescope: Course Exam Data Gradescope [39] is a course application that assists teachers in grading student assignments. This dataset contains 105,218 responses from 6,607 assignments in 2,748 courses and 139 schools. All assignments are instructor-uploaded, fixed-template assignments, with at least 3 questions, with the majority being examinations. We focus on course 102576, randomly chosen. We remove students who did not respond to any questions and round up partial credit. In total, there are 1254 students with 98 items, with missing entries.

PISA 2015: International Assessment The Programme for International Student Assessment (PISA) is an international exam that measures 15-year-old students’ reading, mathematics, and science literacy every three years. It is run by the Organization for Economic Cooperation and Development (OECD). The OECD released anonymized data from PISA ’15 for students from 80 countries and education systems⁶. We focus on the science component. Using IRT to access student performance is part of the pipeline the OECD uses to compute holistic literacy scores for different countries. As part of our processing, we binarize responses, rounding any partial credit to 1. In total, there are 519,334 students and 183 questions. Not every student answers every question as many versions of the computer exam exist.

5. FAST AND ACCURATE INFERENCE

We will show that VIBO is as accurate as HMC and nearly as fast as MLE/EM, making Bayesian IRT a realistic, even preferred, option for modern applications.

5.1 Evaluation

We compare compute cost of VIBO to HMC, EM⁷, and MLE using IRT-2PL by measuring wall-clock run time. For HMC, we limit to drawing 200 samples with 100 warmup steps with no parallelization. For VIBO and MLE, we use the Adam optimizer with a learning rate of 5e-3. We choose to conservatively optimize for 10k iterations to estimate cost.

⁵sharedtask.duolingo.com/2018.html

⁶oecd.org/pisa/data/2015database

⁷We use the popular MIRT package in R for EM with 61 points for numerical integration.

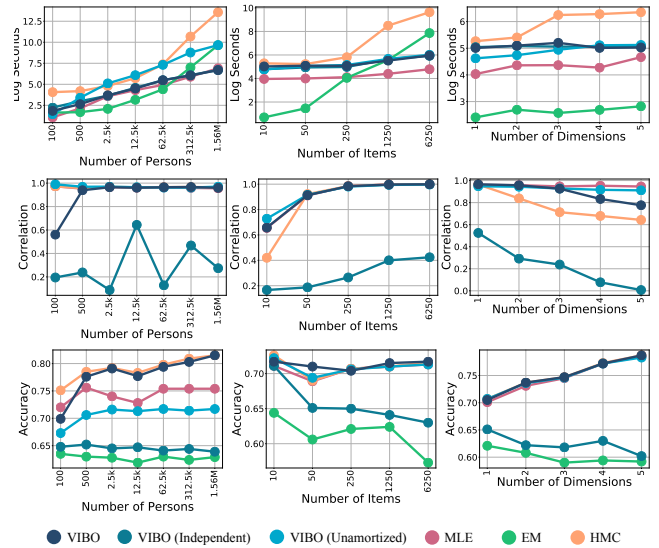


Figure 2: Performance of inference algorithms for IRT for synthetic data, as we vary the number of people, items, and latent ability dimensions. (Top) Computational cost in log-seconds (e.g. 1 log second is about 3 seconds whereas 10 log seconds is 6.1 hours). (Middle) Correlation of inferred ability with true ability (used to generate the data). (Bottom) Accuracy of held-out data imputation.

However, speed only matters assuming good performance. We use three metrics of accuracy: (1) For the synthetic dataset, because we know the true ability, we can measure the expected correlation between it and the inferred ability under each algorithm (with the exception of EM as ability is not inferred). A correlation of 1.0 would indicate perfect inference. (2) The most general metric is the accuracy of imputed missing data. We hold out 10% of the responses, use the inferred ability and item characteristics to generate responses thereby populating missing entries, and compute prediction accuracy for held-out responses. This metric is a good test of “overfitting” to observed responses. (3) In the case of fully Bayesian methods (HMC and VIBO) we can compare posterior predictive statistics [40] to further test uncertainty calibration (which accuracy alone does not capture). Recall that the posterior predictive is defined as:

$$p(\tilde{\mathbf{r}}_{i,1:M} | \mathbf{r}_{i,1:M}) = \mathbb{E}_{p(\mathbf{a}_i, \mathbf{d}_{1:M} | \mathbf{r}_{i,1:M})} [p(\tilde{\mathbf{r}}_{i,1:M} | \mathbf{a}_i, \mathbf{d}_{1:M})]$$

For HMC, we have samples of ability and item characteristics from the true posterior whereas for VIBO, we draw samples from the $q_\phi(\mathbf{a}_i, \mathbf{d}_{1:M} | \mathbf{r}_{i,1:M})$. Given such parameter samples, we can then sample responses. We compare summary statistics of these response samples: the average number of items answered correctly per person and the average number of people who answered each item correctly.

5.2 Synthetic Data Results

With synthetic experiments we are free to vary N and M to extremes to stress test the inference algorithms: first, we range from 100 to 1.5 million people, fixing the number of items to 100 with dimensionality 1; second, we range from 10 to 6k items, fixing 10k people with dimensionality 1; third, we vary the number of latent ability dimensions from 1 to 5, keeping a constant 10k people and 100 items.

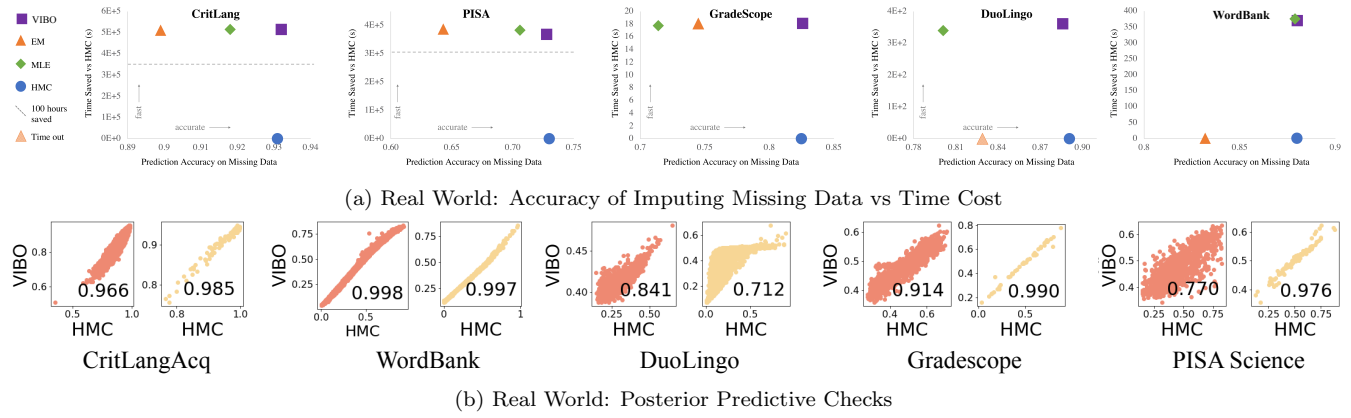


Figure 3: (a) Accuracy of missing data imputation for real world datasets plotted against time saved in seconds compared to using HMC. (b) Samples statistics from the predictive posterior defined using HMC and VIBO. A correlation of 1.0 would be perfect alignment between two inference techniques. Subfigures in red show the average number of items answered correctly for each person. Subfigures in yellow show the average number of people who answered each item correctly.

Fig. 2 shows run-time and performance results for VIBO, MLE, HMC, EM, and two ablations of VIBO (discussed in Sec. 5.4). First, comparing parameter recovery performance (Fig. 2 middle), we see that HMC, MLE and VIBO all recover parameters well. The only notable differences are: VIBO with very few people, and HMC and (to a lesser extent) VIBO in high dimensions. The former is because the amortized posterior approximation requires a sufficiently large dataset (around 500 people) to constrain its parameters. The latter is a simple effect of the scaling of variance for sample-based estimates as dimensionality increases (we fixed the number of samples used, to ease speed comparisons).

Turning to the ability to predict missing data (Fig. 2 bottom) we see that VIBO performs equally well to HMC, except in the case of very few people (again, discussed below). (Note that the latent dimensionality does not adversely affect VIBO or HMC for missing data prediction, because the variance is marginalized away.) MLE also performs well as we scale number of items and latent ability dimensions, but is less able to benefit from more people. EM on the other hand provides much worse missing data prediction in all cases.

Finally if we examine the speed of inference (Fig. 2 top), VIBO is only slightly slower than MLE, both of which are orders of magnitude faster than HMC. For instance, with 1.56 million people, HMC takes 217 hours whereas VIBO takes 800 seconds. Similarly with 6250 items, HMC takes 4.3 hours whereas VIBO takes 385 seconds. EM is the fastest for low to medium sized datasets, though its lower accuracy makes this a dubious victory. Furthermore, EM does not scale as well as VIBO to large datasets.

5.3 Real World Data Results

We next apply VIBO to real world datasets in cognitive science and education. Fig. 3(a) plots the accuracy of imputing missing data against the time saved vs HMC (the most expensive inference algorithm) for five large-scale datasets. Points in the upper right corner are more desirable as they are more accurate and faster. The dotted line represents 100 hours saved compared to HMC.

From Fig. 3(a), we find many of the same patterns as we observed in the synthetic experiments. Running HMC on CritLangAcq or PISA takes roughly 120 hours whereas VIBO takes 50 minutes for CritLangAcq and 5 hours for PISA, the latter being more expensive because of computation required for missing data. In comparison, EM is at times faster than VIBO (e.g. Gradescope, PISA) and at times slower. With respect to accuracy, VIBO and HMC are again identical, outperforming EM by up to 8% in missing data imputation. Interestingly, we find the “overfitting” of MLE to be more pronounced here. If we focus on DuoLingo and Gradescope, the two datasets with pre-existing large portions of missing values, MLE is surpassed by EM, with VIBO achieving accuracies 10% higher.

Another way of exploring a model’s ability to explain data, for fully Bayesian models, is posterior predictive checks. Fig. 3(b) shows posterior predictive checks comparing VIBO and HMC. We find that the two algorithms strongly agree about the average number of correct people and items in all datasets. The only systematic deviations occur with DuoLingo: it is possible that this is a case where a more expressive posterior approximation would be useful in VIBO, since the number of items is greater than the number of people.

5.4 Ablation Studies

We compared VIBO to simpler variants that either do not amortize the posterior or do so with independent distributions of ability and item parameters. These correspond to different variational families, \mathcal{Q} to choose q from:

- VIBO (Independent): We consider the decomposition $q(\mathbf{a}_i, \mathbf{d}_{1:M} | \mathbf{r}_{i,1:M}) = q(\mathbf{a}_i | \mathbf{r}_{i,1:M}) q(\mathbf{d}_{1:M})$ which treats ability and item characteristics as independent.
- VIBO (Unamortized): We consider $q(\mathbf{a}_i, \mathbf{d}_{1:M} | \mathbf{r}_{i,1:M}) = q_{\psi(\mathbf{r}_{i,1:M})}(\mathbf{a}_i) q(\mathbf{d}_{1:M})$, which learns separate posteriors for each \mathbf{a}_i , without parameter sharing. Recall the subscripts $\psi(\mathbf{r}_{i,1:M})$ indicate a separate variational posterior for each unique set of responses.

If we compare unamortized to amortized VIBO in Fig. 2 (top), we see an important efficiency difference. The number

of parameters for the unamortized version scales with the number of people; the speed shows a corresponding impact, with the amortized version becoming an order of magnitude faster than the unamortized one. In general, amortized inference is much cheaper, especially in circumstances in which the number of possible response vectors $\mathbf{r}_{1:M}$ is very large (e.g. 2^{95} for CritLangAcq). Comparing amortized VIBO to the un-amortized equivalent, Table 2 compares the wall clock time (sec.) for the 5 real world datasets. While VIBO is comparable to MLE and EM (Fig. 3a), unamortized VIBO is 2 to 15 times more expensive.

Exploring accuracy in Fig. 2 (bottom), we see that the un-amortized variant is significantly less accurate at predicting missing data. This can be attributed to overfitting to observed responses. With 100 items, there are 2^{100} possible responses from every person, meaning that even large datasets only cover a small portion of the full set. With amortization, overfitting is more difficult as the deterministic mapping f_ϕ is not hardcoded to a single response vector. Without amortization, since we learn a variational posterior for every observed response vector, we may not generalize to new response vectors. Unamortized VIBO is thus much more sensitive to missing data as it does not get to observed the entire response. We can see evidence of this as unamortized VIBO is superior to amortized VIBO at parameter recovery, Fig. 2 (middle), where no data is hidden from the model; compare this to missing data imputation, where unamortized VIBO appears inferior: because ability estimates do not share parameters, those with missing data are less constrained yielding poorer predictive performance.

Finally, when there are very few people (100) unamortized VIBO and HMC are better at recovering parameters (Fig. 2 middle) than amortized VIBO. This can be explained by amortization: to train an effective regressor f_ϕ requires a minimum amount of data. With too few responses, the amortization gap will be very large, leading to poor inference. Under scarce data we would thus recommend using HMC, which is fast enough and most accurate.

Table 2: Time Costs with and without Amortization

DATASET	AMORTIZED (SEC.)	UN-AMORTIZED (SEC.)
CRITLANGACQ	2.8K	43.2K
WORDBANK	176.4	657.1
DUOLINGO	429.9	717.9
GRADESCOPE	114.5	511.1
PISA	25.2K	125.8K

The above suggests that amortization is important when dealing with moderate to large datasets. Turning to the structure of the amortized posteriors, we note that the factorization we chose in Thm. 3.1 is only one of many. Specifically, we could make the simpler assumption of independence between ability and item characteristics given responses in our variational posteriors: VIBO (Independent). Such a factorization would be simpler and faster due to less gradient computation. However, in our synthetic experiments (in which we know the true ability and item features), we found the independence assumption to produce very poor results: recovered ability and item characteristics had less than 0.1 correlation with the true parameters. Meanwhile

the factorization we posed in Thm. 3.1 consistently produced above 0.9 correlation. Thus, the insight to decompose $q(\mathbf{a}_i, \mathbf{d}_{1:M} | \mathbf{r}_{i,1:M}) = q(\mathbf{a}_i | \mathbf{d}_{1:M}, \mathbf{r}_{i,1:M}) q(\mathbf{d}_{1:M} | \mathbf{r}_{i,1:M})$ instead of assuming independence is a critical one. (This point is also supported theoretically by research on faithful inversions of graphical models [43].)

6. DEEP ITEM RESPONSE THEORY

We have found VIBO to be fast and accurate for inference in 2PL IRT, matching HMC in accuracy and EM in speed. This classic IRT model is a surprisingly good model for item responses despite its simplicity. Yet it makes strong assumptions about the interaction of factors, which may not capture the nuances of human cognition. With the advent of much larger data sets we have the opportunity to explore corrections to classic IRT models, by introducing more flexible non-linearities. As described above, a virtue of VI is the possibility of learning aspects of the generative model by optimizing the inference objective. We next explore several ways to incorporate learnable non-linearities in IRT, using the modern machinery of deep learning.

6.1 Nonlinear Generalizations of IRT

We have assumed thus far that $p(\mathbf{r}_{i,1:M} | \mathbf{a}_i, \mathbf{d}_{1:M})$ is a fixed IRT model defining the probability of correct response to each item. We now consider three different alternatives with varying levels of expressivity that help define a class of more powerful nonlinear IRT.

Learning a Linking Function We replace the logistic function in standard IRT with a nonlinear linking function. As such, it preserves the linear relationships between items and people. We call this VIBO (Link). For person i and item j , the 2PL-Link generative model is:

$$p(r_{ij} | \mathbf{a}_i, \mathbf{d}_j) = f_\theta(-\mathbf{a}_i^T \mathbf{k}_j - \mathbf{d}_j) \quad (16)$$

where f_θ is a one-dimensional nonlinear function followed by a sigmoid to constrain the output to be within $[0, 1]$. In practice, we parameterize f_θ as a multilayer perceptron (MLP) with three layers of 64 hidden nodes with ELU nonlinearities.

Learning a Neural Network Here, we no longer preserve the linear relationships between items and people and instead feed the ability and item characteristics directly into a neural network, which will combine the inputs nonlinearly. We call this version VIBO (Deep). For person i and item j , the Deep generative model is:

$$p(r_{ij} | \mathbf{a}_i, \mathbf{d}_j) = f_\theta(\mathbf{a}_i, \mathbf{d}_j) \quad (17)$$

where again f_θ includes a Sigmoid function at the end to preserve the correct output signatures. This is an even more expressive model than VIBO (Link). In practice, we parameterize f_θ as three MLPs, each with 3 layers of 64 nodes and ELU nonlinearities. The first MLP maps ability to a real vector; the second maps item characteristics to a real vector. These two hidden vectors are concatenated and given to the final MLP, which predicts response.

Learning a Residual Correction Although clearly a powerful model, we might fear that VIBO (Deep) becomes too uninterpretable. So, for the third and final nonlinear model, we use the standard IRT but add a nonlinear residual

component that can correct for any inaccuracies. We call this version VIBO (Residual). For person i and item j , the 2PL-Residual generative model is:

$$p(r_{ij}|\mathbf{a}_i, \mathbf{k}_j, d_j) = \frac{1}{1 + e^{-\mathbf{a}_i^T \mathbf{k}_j - d_j + f_\theta(\mathbf{a}_i, \mathbf{k}_j, d_j)}} \quad (18)$$

During optimization, we initialize the weights of the residual network to 0, thus ensuring its initial output is 0. This encourages the model to stay close to IRT, using the residual only when necessary. We use the same architectures for the residual component as in VIBO (Deep).

6.2 Nonlinear IRT Evaluation

A generative model explains the data better when it assigns observations higher probability. We thus evaluate generative models by estimating the log marginal likelihood $\log p(\mathbf{r}_{1:N,1:M})$ of the training dataset. A higher number (closer to 0) is better. For a single person, the log marginal likelihood of his or her M responses can be computed as:

$$\log p(\mathbf{r}_{i,1:M}) \approx \log \mathbb{E}_{q_\phi(\mathbf{a}_i, \mathbf{d}_{1:M} | \mathbf{r}_{i,1:M})} \left[\frac{p_\theta(\mathbf{r}_{i,1:M}, \mathbf{a}_i, \mathbf{d}_{1:M})}{q_\phi(\mathbf{a}_i, \mathbf{d}_{1:M} | \mathbf{r}_{i,1:M})} \right] \quad (19)$$

We use 1000 samples to estimate Eq. 19. We also measure accuracy on missing data imputation as we did in Sec. 5. A more powerful generative model, that is more descriptive of the data, should be better at filling in missing values.

6.3 Nonlinear IRT Results

The top half of Table 3 compares the log likelihoods of observed data whereas the bottom half of Table 3 compares the accuracy of imputing missing data. We include VIBO inference with classical IRT-1PL and IRT-2PL generative models as baselines. We find a consistent trend: the more powerful generative models achieve a higher log likelihood (closer to 0) and a higher accuracy. In particular, we find very large increases in log likelihood moving from IRT to Link, spanning 100 to 500 log points depending on the dataset. Further, from Link to Deep and Residual, we find another increase of 100 to 200 log points. In some cases, we find Residual to outperform Deep, though the two are equally parameterized, suggesting that initialization with IRT can find better local optima. These gains in log likelihood translate to a consistent 1 to 2% increase in held-out accuracy for Link/Deep/Residual over IRT. This suggests that the datasets are large enough to use the added model flexibility appropriately, rather than overfitting to the data.

We also compare our deep generative IRT models with the purely deep learning approach called Deep-IRT [47] (see Sec. 8), that does not model posterior uncertainty. Unlike traditional IRT models, Deep-IRT was built for knowledge tracing and assumed sequential responses. To make our datasets amenable to Deep-IRT, we assume an ordering of responses from $j = 1$ to $j = M$. As shown in Table 3, our models outperform Deep-IRT in all 5 datasets by as much as 30% in missing data imputation (e.g. WordBank).

6.4 Interpreting the Linking Function

With nonlinear models, we face an unfortunate tradeoff between interpretability and expressivity. In domains like education, practitioners greatly value the interpretability of IRT where predictions can be directly attributed to ability

or item features. With VIBO (Deep), our most expressive model, predictions use a neural network, making it hard to understand the interactions between people and items.

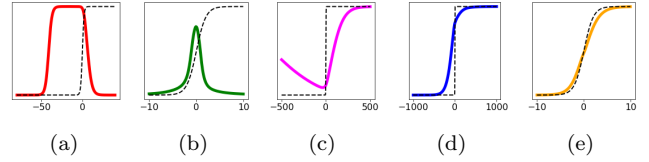


Figure 4: Learned link functions for (a) CritLangAcq, (b) WordBank, (c) DuoLingo, (d) Gradescope, and (e) PISA. The dotted black line shows the default logistic function.

Fortunately, with VIBO (Link), we can maintain a degree of interpretability along with power. The “Link” generative model is identical to IRT, only differing in the linking function (i.e. item response function). Each subfigure in Fig. 4 shows the learned response function for one of the real world datasets; the dotted black line represents the best standard linking function, a sigmoid. We find three classes of linking functions: (1) for Gradescope and PISA, the learned function stays near a Sigmoid. (2) For WordBank and CritLangAcq, the response function closely resembles an unfolding model [27, 2], which encodes a more nuanced interaction between ability and item characteristics: higher scores are related to higher ability only if the ability and item characteristics are “nearby” in latent space. (3) For DuoLingo, we find a piecewise function that resembles a sigmoid for positive values and a negative linear function for negative values. In cases (2) and (3) we find much greater differences in log likelihood between VIBO (IRT) and VIBO (Link). See Table 3. For DuoLingo, VIBO (Link) matches the log density of more expressive models, suggesting that most of the benefit of nonlinearity is exactly in this unusual linking function.

7. POLYTOMOUS RESPONSES

Thus far, we have been working only with response data collapsed into binary correct/incorrect responses. However, many questionnaires and examinations are not binary: responses can be multiple choice (e.g. Likert scale) or even real valued (e.g. 92% on a course test). Having posed IRT as a generative model, we have prescribed a Bernoulli distribution over the i -th person’s response to the j -th item. Yet nothing prevents us from choosing a different distribution, such as Categorical for multiple choice or Normal for real-values. The DuoLingo dataset contains partial credit, computed as a fraction of times an individual gets a word correct. A more granular treatment of these polytomous values should yield a more faithful model that can better capture the differences between people. We thus modeled the DuoLingo data using for $p(\mathbf{r}_{i,1:M}|\mathbf{a}_i, \mathbf{d}_{1:M})$ a (truncated) Normal distribution over responses with fixed variance. Table 4 show the log densities: we again observe large improvements from nonlinear models.

Item Response Theory can in this way be extended to work of all kinds (imagine students writing text, drawing pictures, or even coding), encouraging educators to assign open-ended work without having to give up proper tools of assessment.

8. RELATED WORK

Table 3: Log Likelihoods and Missing Data Imputation for Deep Generative IRT Models

DATASET	DEEP IRT	VIBO (IRT-1PL)	VIBO (IRT-2PL)	VIBO (LINK-2PL)	VIBO (DEEP-2PL)	VIBO (RES.-2PL)
CRITLANGACQ	-	-11249.8 ± 7.6	-10224.0 ± 7.1	-9590.3 ± 2.1	-9311.2 ± 5.1	- 9254.1 ± 4.8
WORDBANK	-	-17047.2 ± 4.3	-5882.5 ± 0.8	-5268.0 ± 7.0	- 4658.4 ± 3.9	-4681.4 ± 2.2
DUOLINGO	-	-2833.3 ± 0.7	-2488.3 ± 1.4	-1833.9 ± 0.3	-1834.2 ± 1.3	- 1745.4 ± 4.7
GRADESCOPE	-	-1090.7 ± 2.9	-876.7 ± 3.5	-750.8 ± 0.1	- 705.1 ± 0.5	-715.3 ± 2.7
PISA	-	-13104.2 ± 5.1	-6169.5 ± 4.8	-6120.1 ± 1.3	-6030.2 ± 3.3	- 5807.3 ± 4.2
CRITLANGACQ	0.934	0.927	0.932	0.945	0.948	0.947
WORDBANK	0.681	0.876	0.880	0.888	0.889	0.889
DUOLINGO	0.884	0.880	0.886	0.891	0.897	0.894
GRADESCOPE	0.813	0.820	0.826	0.840	0.847	0.848
PISA	0.524	0.723	0.728	0.718	0.744	0.739

Table 4: DuoLingo with Polytomous Responses

INF. ALG.	TRAIN	TEST
VIBO (IRT)	-22038.07	-21582.03
VIBO (LINK)	-17293.35	-16588.06
VIBO (DEEP)	- 15349.84	- 14972.66
VIBO (RES.)	-15350.66	-14996.27

We described above a variety of methods for parameter estimation in IRT such as MLE, EM, and MCMC. The benefits and drawbacks of these methods are well-documented [26], so we need not discuss them here. Instead, we focus specifically on methods that utilize deep neural networks or variational inference to estimate IRT parameters.

While variational inference has been suggested as a promising alternative to other inference approaches for IRT [26], there has been surprisingly little work in this area. In an exploration of Bayesian prior choice for IRT estimation, Natesan et al. [31] posed a variational approximation to the posterior:

$$p(\mathbf{a}_i, \mathbf{d}_j | r_{i,j}) \approx q_\phi(\mathbf{a}_i, \mathbf{d}_j) = q_\phi(\mathbf{a}_i)q_\phi(\mathbf{d}_j) \quad (20)$$

This is an unamortized and independent posterior family, unlike VIBO. As we noted in Sec. 5.4, both amortization and dependence of ability on items were crucial for our results.

We are aware of two approaches that incorporate deep neural networks into Item Response Theory: Deep-IRT [46] and DIRT [8]. Deep-IRT is a modification of the Dynamic Key-Value Memory Network (DKVMN) [47] that treats data as longitudinal, processing items one-at-a-time using a recurrent architecture. Deep-IRT produces point estimates of ability and item difficulty at each time step, which are then passed into a 1PL IRT function to produce the probability of answering the item correctly. The main difference between DIRT and Deep-IRT is the choice of neural network: instead of the DKVMN, DIRT uses an LSTM with attention [41]. In our experiments, we compare our approach to Deep-IRT and find that we outperform it by up to 30% on the accuracy of missing response imputation. On the other hand, our models do not capture the longitudinal aspect of response data. Combining the two approaches would be natural.

Lastly, Curi et al. [9] used a VAE to estimate IRT parameters in a 28-question synthetic dataset. However, this approach modeled ability as the only unknown variable, ignoring items. Our analogue to the VAE builds on the IRT graphical model,

incorporating both ability and item characteristics in a principled manner. This could explain why Curi et. al. report the VAE requiring substantially more data to recover the true parameters when compared to MCMC whereas we find comparable data-efficiency between VIBO and MCMC.

9. BROADER IMPACT

We briefly emphasize the broader impact of efficient IRT in the context of education. Firstly, one of the many difficulties of accurately estimating student ability is cost: attempting to use MCMC on the order magnitude required by large entities like MOOCs, local and national governments, and international organizations is impossible. However with VIBO, doing so is already possible, as shown by the PISA results. Second, efficient IRT is an important and necessary step to encourage the development of more complex models of student cognition and response. Namely, it will at least enable faster research and iterative testing on real world data.

10. CONCLUSION

Item Response Theory is a paradigm for reasoning about the scoring of tests, surveys, and similar measurement instruments. Notably, the theory plays an important role in education, medicine, and psychology. Inferring ability and item characteristics poses a technical challenge: balancing efficiency against accuracy. In this paper we have found that variational inference provides a potential solution, running orders of magnitude faster than MCMC algorithms while matching their state-of-the-art accuracy.

Many directions for future work suggest themselves. First, further gains in speed and accuracy could be found by exploring more or less complex families of posterior approximation. Second, more work is needed to understand deep generative IRT models and determine the most appropriate tradeoff between expressivity and interpretability. For instance, we found significant improvements from a learned linking function, yet in some applications monotonicity may be judged important to maintain – greater ability, for instance, should correspond to greater chance of success. Finally, VIBO should enable more coherent, fully Bayesian, exploration of very large and important datasets, such as PISA [13].

Recent advances within AI combined with new massive datasets have enabled advances in many domains. We have given an example of this fruitful interaction for understanding humans based on their answers to questions.

11. ACKNOWLEDGMENTS

We thank Ben Stenhaus for helpful discussions. This work was supported in part by DARPA under agreement FA8650-19-C-7923, by the Office of Naval Research grant ONR MURI N00014-16-1-2007, and in part by a gift from an anonymous donor. Additionally, MW is supported by NSF GRFP.

12. REFERENCES

- [1] T. A. Ackerman. Using multidimensional item response theory to understand what items and tests are measuring. *Applied Measurement in Education*, 7(4):255–278, 1994.
- [2] D. Andrich and G. Luo. A hyperbolic cosine latent trait model for unfolding dichotomous single-stimulus responses. *Applied Psychological Measurement*, 17(3):253–276, 1993.
- [3] A. A. Béguin and C. A. Glas. Mcmc estimation and some model-fit analysis of multidimensional irt models. *Psychometrika*, 66(4):541–561, 2001.
- [4] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.
- [5] R. D. Bock and M. Aitkin. Marginal maximum likelihood estimation of item parameters: Application of an em algorithm. *Psychometrika*, 46(4):443–459, 1981.
- [6] R. D. Bock, R. Gibbons, and E. Muraki. Full-information item factor analysis. *Applied Psychological Measurement*, 12(3):261–280, 1988.
- [7] M. Braginsky, D. Yurovsky, V. A. Marchman, and M. C. Frank. Developmental changes in the relationship between grammar and the lexicon. In *CogSci*, pages 256–261, 2015.
- [8] S. Cheng, Q. Liu, E. Chen, Z. Huang, Z. Huang, Y. Chen, H. Ma, and G. Hu. Dirt: Deep learning enhanced item response theory for cognitive diagnosis. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2397–2400, 2019.
- [9] M. Curi, G. A. Converse, J. Hajewski, and S. Oliveira. Interpretable variational autoencoders for cognitive models. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.
- [10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.
- [11] F. Y. Edgeworth. The statistics of examinations. *Journal of the Royal Statistical Society*, 51(3):599–635, 1888.
- [12] S. E. Embretson and S. P. Reise. *Item response theory*. Psychology Press, 2013.
- [13] O. for Economic Co-operation and D. (OECD). Pisa 2015 database. 2016.
- [14] M. C. Frank, M. Braginsky, D. Yurovsky, and V. A. Marchman. Wordbank: An open repository for developmental vocabulary data. *Journal of child language*, 44(3):677–694, 2017.
- [15] A. E. Gelfand and A. F. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.
- [16] S. Gershman and N. Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014.
- [17] S. J. Haberman. Maximum likelihood estimates in exponential response models. *The Annals of Statistics*, pages 815–841, 1977.
- [18] W. Harlen. The assessment of scientific literacy in the oecd/pisa project. 2001.
- [19] J. K. Hartshorne, J. B. Tenenbaum, and S. Pinker. A critical period for second language acquisition: Evidence from 2/3 million english speakers. *Cognition*, 177:263–277, 2018.
- [20] M. R. Harwell, F. B. Baker, and M. Zwarts. Item parameter estimation via marginal maximum likelihood and an em algorithm: A didactic. *Journal of Educational Statistics*, 13(3):243–271, 1988.
- [21] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.
- [22] G. E. Hinton. Products of experts. 1999.
- [23] M. D. Hoffman and A. Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.
- [24] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.
- [25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [26] W. J. v. d. Linden. *Handbook of Item Response Theory: Volume 2: Statistical Tools*. CRC Press, 2017.
- [27] C.-W. Liu and R. P. Chalmers. Fitting item response unfolding models to likert-scale data using mirt in r. *PloS one*, 13(5), 2018.
- [28] L. Magdalena, E. Haman, S. A. Lotem, B. Etenkowski, F. Southwood, D. Andjelkovic, E. Bloom, T. Boerma, S. Chiat, P. E. de Abreu, et al. Ratings of age of acquisition of 299 words across 25 languages: Is there a cross-linguistic order of words? *Behavior Research Methods (print Edition)*, 48(3):1154–1177, 2016.
- [29] R. P. McDonald. A basis for multidimensional item response theory. *Applied Psychological Measurement*, 24(2):99–114, 2000.
- [30] A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.
- [31] P. Natesan, R. Nandakumar, T. Minka, and J. D. Rubright. Bayesian prior choice in irt estimation using mcmc and variational bayes. *Frontiers in psychology*, 7:1422, 2016.
- [32] R. M. Neal. An improved acceptance procedure for the hybrid monte carlo algorithm. *Journal of Computational Physics*, 111(1):194–203, 1994.
- [33] R. M. Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.
- [34] G. Rasch. Studies in mathematical psychology: I. probabilistic models for some intelligence and attainment tests. 1960.
- [35] D. Ravitch. *National standards in American education*: 1990.

A citizen's guide. ERIC, 1995.

- [36] M. D. Reckase. Multidimensional item response theory models. In *Multidimensional item response theory*, pages 79–112. Springer, 2009.
- [37] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.
- [38] B. Settles, C. Brust, E. Gustafson, M. Hagiwara, and N. Madnani. Second language acquisition modeling. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 56–65, 2018.
- [39] A. Singh, S. Karayev, K. Gutowski, and P. Abbeel. Gradescope: a fast, flexible, and fair system for scalable assessment of handwritten work. In *Proceedings of the fourth (2017) acm conference on learning@ scale*, pages 81–88. ACM, 2017.
- [40] S. Sinharay, M. S. Johnson, and H. S. Stern. Posterior predictive assessment of item response theory models. *Applied Psychological Measurement*, 30(4):298–321, 2006.
- [41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [42] M. J. Wainwright, M. I. Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.
- [43] S. Webb, A. Golinski, R. Zinkov, S. Narayanaswamy, T. Rainforth, Y. W. Teh, and F. Wood. Faithful inversion of generative models for effective amortized inference. In *Advances in Neural Information Processing Systems*, pages 3070–3080, 2018.
- [44] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3–4):229–256, 1992.
- [45] M. Wu and N. Goodman. Multimodal generative models for scalable weakly-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5575–5585, 2018.
- [46] C.-K. Yeung. Deep-irt: Make deep learning based knowledge tracing explainable using item response theory. *arXiv preprint arXiv:1904.11738*, 2019.
- [47] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pages 765–774, 2017.

Student Subtyping via EM-Inverse Reinforcement Learning

Xi Yang¹, Guojing Zhou¹, Michelle Taub², Roger Azevedo², Min Chi¹

¹ North Carolina State University, Raleigh, NC 27606, USA
{yxi2, gzhou3, mchi}@ncsu.edu

² University of Central Florida, Orlando, FL 32816, USA
{michelle.taub, roger.azevedo}@ucf.edu

ABSTRACT

In the learning sciences, heterogeneity among students usually leads to different learning strategies or patterns and may require different types of instructional interventions. Therefore, it is important to investigate student subtyping, which is to group students into subtypes based on their learning patterns. Subtyping from complex student learning processes is often challenging because of the information heterogeneity and temporal dynamics. Various inverse reinforcement learning (IRL) algorithms have been successfully employed in many domains for inducing policies from the trajectories and recently has been applied for analyzing students' temporal logs to identify their domain knowledge patterns. IRL was originally designed to model the data by assuming that all trajectories have a *single* pattern or strategy. Due to the heterogeneity among students, their strategies can vary greatly and the design of traditional IRL may lead to suboptimal performance. In this paper, we applied a novel expectation-maximization IRL (EM-IRL) to extract heterogeneous learning strategies from sequential data collected from three simulation environments and real-world longitudinal students' logs. Experiments on simulation environments showed that EM-IRL can successfully identify different policies from the heterogeneous sequences with different strategies. Furthermore, experimental results from our educational dataset showed that EM-IRL can be used to obtain different student subtypes: a *"learning-oriented"* subtype who learned the material as much as possible regardless of the time in that they spent significantly more time than the other two subtypes and learned significantly; an *"efficient-oriented"* subtype who learned efficiently in that they not only learned significantly but also spent less time than the first subtype; a *"no learning"* subtype who spent less amount of time than first subtype and failed to learn.

Keywords

Subtyping, learning progression modeling, Student strategy, Inverse reinforcement learning

1. INTRODUCTION

With the rapid development of educational technologies, longitudinal students' learning progression trajectories are readily available. It is often challenging to analyze large-scale heterogeneous progression trajectories to infer high-level information embedded in student subgroups. This challenge motivates the development of student modeling [1, 2, 3, 4] and instructional intervention [5, 6, 7, 8].

Student subtyping, which seeks student groups with similar learning progression trajectories, is crucial to address the heterogeneity in the students, which ultimately leads to personalized instruction where students are provided with interventions tailored to their unique learning status. Student subtyping facilitates the investigation of different types of pedagogical strategies. From the data mining perspective, student subtyping is posed as an unsupervised clustering task of grouping students according to their historical records. Since these records are longitudinal and interrelated, it is important to capture the dependencies among the elements of the recorded sequence to learn more effective and robust representations, which can be utilized in the clustering stage to obtain the student subgroups.

This work aims at investigating *student subtyping* based on their pedagogical strategies, which can be seen as a process of self-regulated learning [9, 10, 11, 12, 13] by setting one's learning goals and ensuring the goals to be attained. Specifically, we focus on students' pedagogical decision-making strategies during their interactions with an intelligent tutoring system (ITS) to learn the probability. In this ITS, once a problem is presented, the students will decide whether they want the ITS *to tell* them how to solve the next problem or complete the next step, by presenting a worked example, or they want the ITS *to elicit* the next problem or take the next step themselves, by requiring problem solving. When making pedagogical decisions, the students have to self-regulate their own learning process which may change the learning outcomes even though the instructional content is controlled. We believe that students' pedagogical strategies are closely related to metacognition, i.e., the processes involved in thinking about thinking [14].

Reinforcement learning (RL) offers one of the most promising approaches to induce effective pedagogical strategies directly from data. A number of researchers have studied applying RL to improve the effectiveness of ITSs, e.g. [15, 7, 16, 17, 18, 19, 20, 8, 21, 22], and much of the prior work fo-

Xi Yang, Guojing Zhou, Michelle Taub, Roger Azevedo and Min Chi "Student Subtyping via EM-Inverse Reinforcement Learning" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 269 - 279

cused on inducing effective policies that determine the best action for the *ITS* to take in any given situation so as to maximize a cumulative reward, which is often the student learning gain. On the other hand, in this work, our goal is to infer *students'* pedagogical strategies based on their behaviors and decisions while interacting with the ITS.

To do so, we applied *inverse reinforcement learning (IRL)*. Unlike RL, where the reward function is explicitly given as input, IRL takes a bunch of trajectories as input and from which a reward function will be inferred. Given this inferred reward function, the RL can be further deployed to induce the decision-making policy. Since the students' decisions are generally made based on a trade-off among various complex factors, e.g., time, learning gain, difficulty of problems, etc., merely taking the learning gain as the reward cannot reflect the actual decision-making patterns. As a result, we employed IRL to learn students' strategies based on their behavioural data. Recently, IRL has been widely employed in various domains to understand how decisions are made in the given trajectories [23, 24]. Specifically, it has been employed in educational domains to analyze students' temporal log data to identify their domain knowledge patterns [25, 26]. However, IRL was originally designed to model the data by assuming that all trajectories share a *single* pattern or strategy. Considering the heterogeneity among students, their pedagogical strategies can vary greatly and the design of traditional IRL may lead to suboptimal performance. Though we can apply IRL individually for each student, it will forfeit our goal of revealing some general and meaningful patterns from students' trajectories in consideration of the heterogeneity among subgroups of students.

We employed a novel expectation-maximization IRL (EM-IRL) algorithm [27] to model the heterogeneity among student subtypes by assuming that different student subtypes have different pedagogical strategies and students within each subtype share the same strategy. The EM-IRL would recursively cluster students into different subgroups and induce a policy for each group by IRL until both clusters and policies get converged. In the original EM-IRL work, it requires the number of clusters to be pre-defined [27]. However, when applying it to student subtyping in education, it is often hard to figure out beforehand how many types of strategies are involved in students' trajectories. Therefore, we embedded the original EM-IRL into a general framework which can automatically determine the optimal number of clusters from the data.

In this work, we evaluated our general framework on three simulation environments: Grid World, Highway, and Mountain Car, and on real-world longitudinal students' logs collected from an ITS. Our results in three simulation environments showed that EM-IRL could accurately cluster the data with different decision-making strategies. In addition, the experimental results showed that EM-IRL could be easily employed to obtain the student subtypes. Specifically, we got three student subtypes: a "*learning-oriented*" subtype who try to learn the material as much as possible regardless of the time spent and they learned significantly from pre- to post-test; an "*efficient-oriented*" subtype who learn efficiently in that they not only learned significantly but also spent significantly less time than the first subtype; a "*no*

learning" subtype who spent the less time and failed to learn. The clustering results suggested the potential of targeting the students who are not using effective pedagogical strategies, adapting the interventions, and offering the students effective pedagogical skill training through the ITS.

The remaining parts are organized as follows. In Section 2, related works are reviewed. Section 3 presents the methods, including the RL, IRL, and EM-IRL. Section 4 displays preliminary results we got in three simulation environments. Section 5 details data collected from the ITS. In Section 6, we discuss the experimental setup for EM-IRL and some other clustering methods. Section 7 presents the experimental results. Finally, Section 8 summarizes the paper.

2. RELATED WORKS

2.1 Students' Subtyping

Previous research has widely explored modeling of student subtyping to assist teachers in providing more targeted interventions at the right time. Generally, student subtyping was analyzed via unsupervised clustering methods. For example, Lopez et al. employed an expectation maximisation clustering method to determine if the students' participation in course Moodle forum could be a good predictor of the final marks [28]. Durairaj and Vijitha applied K-means clustering to predict the pass/fail percentage of the students who appeared for a particular examination [29]. Khalil and Ebner clustered the students into appropriate categories based on their level of engagement [30], so that the teachers could increase retention and improve interventions for specific sub-population. All of these methods were based on the static data, without considering the dynamic properties during learning.

With the rapid development of e-learning, an increasing amount of sequential data was collected via ITSs. In general, the clustering methods to handle sequential data could be generalized into three categories: proximity-based, feature-based, and model-based [31]. More specifically, proximity-based methods measures the similarity between the pair-wise data via the distance calculated by the longest common subsequence, dynamic time warping, etc. For example, Shen and Chi proposed a temporal clustering framework which measured pair-wise distance between the students by dynamic time warping and then clustered them by hierarchical clustering [32]. Their method identified some distinctive patterns among the clusters, which could provide benefits to the personalized learning. Feature-based approaches would first compress the sequential data to be static, then the clustering methods taking static data as input could be further employed. For example, in [33] and [34], the authors aggregated the students' activities to a feature vector and then applied K-means clustering to recognize learner groups in exploratory learning environments. In the model-based methods, the similarity of two data could be calculated based on the likelihood of one of them given the model derived from the other. For example, Li and Yoo proposed to use a Markov chain based clustering methodology to model the students' online learning behaviors collected during the learning process for more effective and adaptive teaching [31]. Additionally, Kock and Paramythis proposed a method combining K-means clustering with dis-

crete Markov models to identify new, semantically meaningful problem-solving styles of the learners [35].

2.2 Students' Pedagogical Strategies

A number of researchers have investigated students' pedagogical decision-making [36, 37, 38, 39, 40, 41]. Previous research has shown that students make pedagogical decisions strategically. For example, Aleven et al. conducted a study to investigate students' hint usage behavior [36]. Results showed that students used the easy-to-apply intelligent help more often than the Glossary. However, students often waited long before asking for a hint. When requesting hints, they often skipped the intermediate hints to reach the bottom-out hint which showed the solution directly. The results suggested that students preferred less effort-taking help (intelligent help and bottom-out hint), and oftentimes, they used the help less than they needed.

Additionally, prior research showed that providing students with pedagogical decision-making assistance could result in better decision-making skills or learning performance. Roll et al. [37] examined the relationship between students' help-seeking patterns and learning performance. They found that asking for help on challenging steps was generally productive while help abusing behaviors were correlated with poor learning. Mitrovic et al. [38] compared three types of decision-making modes: system control, student control, and faded control. Under the faded control, the system selected the problem for the student at the beginning of the training and gave explanations of why the problems should be selected. As the training proceeded, the control was given to the students. Results showed that the faded control group demonstrated improved problem selection skill and achieved better learning gain than the other two groups. Long et al. [39] compared an assistance condition, where problem selection assistance was provided, with standard condition (no assistance). Their results showed the assistance condition achieved significantly better learning performance and better declarative knowledge of a key problem-selection strategy comparing to the standard condition.

2.3 Learning From Demonstrations

Learning from demonstrations [42], also known as imitation learning [43] or apprenticeship learning [44], is a process to reproduce the decision-making behaviors in demonstrated trajectories. Generally, the methods in this area can be categorized into two groups: 1) directly learning a policy as a state-action mapping by parroting the demonstrated behaviors, which is typically done via supervised learning; and 2) inferring rewards from the demonstrations and then applying reinforcement learning (RL) to induce the policy, which is called inverse reinforcement learning (IRL). The latter is generally preferred because the reward is a more robust, succinct, and transferable definition for a task [45]. Specifically, comparing to supervised learning, IRL has higher generalization ability to robustly learn from smaller size trajectories collected from larger state spaces, and the succinctly represented reward function can be handily transferred to other agents in different scenarios.

Based on how the rewards are inferred, existing IRL algorithms can be generalized into two categories: maximum margin-based methods and probabilistic model-based meth-

ods. Specifically, maximum margin-based methods infer rewards by finding a model to maximize the margin between the demonstrated trajectories and other alternative behaviors [44]. However, it is often suffers from the ill-posed issue with non-uniqueness [45], i.e., there can be multiple reward functions to explain the demonstrated behaviors. Probabilistic model-based methods, on the other hand, are able to handle this issue by using probability distributions to introduce preferences over reward functions [46]. In this category, Ramachandran and Amir [47] proposed a Bayesian IRL, which combined prior knowledge and evidence from the demonstrated trajectories to derive a probability distribution over the reward functions. Similarly, Ziebart et al. proposed a maximum entropy IRL which results in the least biased estimation of the reward function [23]. Babes-Vroman et al. [27] proposed a maximum likelihood IRL (MLIRL), which finds the reward function that maximize the probability to observe the demonstrated behaviors. Their experimental results showed that the MLIRL outperformed some other IRL methods, including the linear programming based maximum margin IRL and maximum entropy IRL.

All the above methods assume a *single* reward function for all demonstrations. Some other approaches have been proposed to handle the *multiple* reward functions. Dimitrakakis and Rothkopf [48] proposed a Bayesian multi-task IRL, which learns a reward function for each individual trajectory using the same prior distribution. Choi et al. [49] proposed a method based on nonparametric Bayesian IRL in which the prior of mixing distribution of different rewards was modeled by the Dirichlet process. Babes-Vroman et al. [27] proposed an EM-based framework, which iteratively computes the probabilities that the demonstrations belong to each cluster and updates the cluster-wise rewards based on MLIRL. Considering the efficiency and good performance EM-based method, we adapted it for analysis in this work.

Recently, IRL has been widely applied in various domains. Ziebart et al. [23] employed it in driver route modeling for predicting driving behaviors as well as for route recommendation. Asoh et al. [24] applied IRL to medical records and explored the potential rules in doctors' diagnosis. Of most relevance, IRL also showed effectiveness in educational domain. Rafferty et al. applied IRL in education applications to automatically infer learners' beliefs in an education game [25]. They demonstrated that IRL could recover the participant's beliefs towards how their actions could affect the environment, which indicated the potential to utilize IRL to interpret data from interactive educational environments. Then in another of their work, IRL was further employed to assess learners' mastery of some skills in solving algebraic equations. Based on the learned IRL results, some skills the learners misunderstood could be detected and personalized feedback for improving the skills were further rendered [26].

3. METHOD

3.1 Reinforcement Learning

Markov decision process (MDP) was widely utilized to model the user-system interactions. The central idea behind reinforcement learning (RL) is to transform the problem of inducing effective policies into a computational problem of finding an optimal policy for choosing actions in MDP. An

MDP describes a stochastic control process using a 5-tuple $\langle S, A, T, R, \gamma \rangle$. Taking the pedagogical policy induction as an example, S indicates the learning environment states, which is often represented by student-system interaction features. A denotes the tutor's possible actions, such as elicit or tell. The reward function R is generally assigned as students' learning performance. The transition probability T can be estimated from training data. $\gamma \in [0, 1)$ denotes a discount factor for the future rewards. Given a defined MDP, we can transform our student-system interaction logs into trajectories as: $s_1 \xrightarrow{a_1, r_1} s_2 \xrightarrow{a_2, r_2} \dots s_n \xrightarrow{a_n, r_n} s_{n+1}$. Here $s_i \xrightarrow{a_i, r_i} s_{i+1}$ indicates that at the i^{th} turn, the learning environment was in state s_i ; the tutor executed action a_i and received reward r_i ; then the environment transferred into the state s_{i+1} .

In traditional RL, the reward function R serves as a guidance to praise or punish the agent's behaviors to fulfil a certain task when interacting with the environment. Therefore, it is essential and needs to be elaborately hand-crafted in advance to reflect the task. In ITS, the reward is generally formulated as the students' learning performance, e.g., learning gains, since the intention of tutor's decision-making is to promote students' learning. However, the reward function in students' decision-making is more complex to be determined: students may have various learning patterns, e.g., finishing the process as quick as possible or working hard regardless of the time, which is cumbersome to be manually encoded in a reward function. The different reward functions reflected the different strategies students employed during the training process. Therefore, if student's reward function can be learned in a data-driven manner, we can better understand their pedagogical decision-making strategies.

3.2 Inverse Reinforcement Learning

3.2.1 General IRL

The difficulty of the reward function design triggered the development of the inverse reinforcement learning (IRL). IRL follows a reverse procedure comparing to the traditional RL: in RL, given the reward function, the agent will learn an optimal policy; while in IRL, the trajectories derived from the optimal policy are given, from which the agent will learn the reward function. It can be described as a stochastic control process using a 4-tuple $MDP \setminus R = \langle S, A, T, \gamma \rangle$ where the reward function is missing.

In general framework of IRL, the input is a $MDP \setminus R$ together with some demonstrated trajectories \mathcal{T} . The reward function \mathcal{R}_θ parameterized by θ can be modeled as either a linearly weighted sum of feature values or belonging to a certain distribution. Most of the existing IRL methods follow 3 steps: in step 1, the parameter θ is randomly initialized; in step 2, given the \mathcal{R}_θ , general RL methods can be applied to induce the policy; In step 3, the divergence of the behaviors regarding to the learned policy and the given trajectories is minimized to update the θ . The step 2 and step 3 are repeated until the divergence is reduced to a desired level.

To investigate students' pedagogical strategy, we can feed their decision-making trajectories into the IRL model. Once the reward function is learned, the strategy can be further induced via traditional RL methods. Herein, we compared some most commonly utilized IRL methods including: quadratic programming based maximum margin IRL [44],

General Process of IRL

Input $MDP \setminus R = \langle S, A, T, \gamma \rangle$ and trajectories \mathcal{T}

Output \mathcal{R}_θ

step 1 Initialize the parameter θ in reward function

step 2 Solve the MDP to learn the policy π

step 3 Update the optimization θ to minimize the divergence between \mathcal{T} and behaviors following the π

Repeat step 2 and step 3 until convergence

maximum entropy IRL [23], Bayesian IRL [47], and maximum likelihood IRL (MLIRL) [27] over three online simulation environments (i.e., Grid World, Highway, and Mountain Car). We found MLIRL always outperformed others and it is also most time-efficient. As a result, we take MLIRL for the IRL-based analysis hereinafter.

3.2.2 Maximum Likelihood IRL

To formally define the maximum likelihood IRL, we denote the input N demonstrated trajectories as $\mathcal{T} = \{\xi_1, \dots, \xi_N\}$ and each trajectory is composed of a set of state-action pairs: $\xi_i = \{(s_1, a_1), (s_2, a_2), \dots\}$. The reward function is defined as the linear function of feature vector for state-action pairs: $r_\theta(s, a) = \theta^T \phi(s, a)$. Then the Q-value can be calculated as:

$$Q_\theta(s, a) = \theta^T \phi(s, a) + \gamma \sum_{s'} T(s, a, s') \bigotimes_{a'} Q_\theta(s', a'), \quad (1)$$

$$\text{where } \bigotimes_a Q_\theta(s, a) = \frac{\sum_a Q_\theta(s, a) \exp(\beta Q_\theta(s, a))}{\sum_{a'} \exp(\beta Q_\theta(s, a'))} \quad (2)$$

Eq. 2 shows the Boltzmann exploration. Comparing to standard Bellman equation, it enables the likelihood to be differentiable, thus the objective function can be easier optimized. β represents the degree of confidence and it is set as 0.5 in our experiments. The Boltzmann exploration policy parameterized by θ is:

$$\pi_\theta(s, a) = \frac{\exp(\beta Q_\theta(s, a))}{\sum_{a'} \exp(\beta Q_\theta(s, a'))} \quad (3)$$

Then the log-likelihood of trajectories \mathcal{T} is calculated as:

$$L(\mathcal{T}|\theta) = \log \prod_{i=1}^N \prod_{(s,a) \in \xi_i} \pi_\theta(s, a)^{\omega_i} = \sum_{i=1}^N \sum_{(s,a) \in \xi_i} \omega_i \log \pi_\theta(s, a) \quad (4)$$

Herein, ω_i denote the weight for ξ_i , which can be estimated by its frequency of the occurrence. By maximizing the Eq. 4, the parameter θ enables the trajectories \mathcal{T} to have highest probability to be observed given the reward function \mathcal{R}_θ . Once the reward function is learned, the strategy followed by \mathcal{T} can be further induced by any RL method, e.g., policy iteration that we employed in this work.

In general, IRL methods assume the reward function to be unique for all input trajectories. However, it is often the case that the trajectories are heterogeneous and have various reward functions. For example, in ITS, students' decision-making behaviors can have different patterns which cannot be easily captured by a single IRL model. As a result, a model suitable for multiple reward functions is favored.

Algorithm: MLIRL

Input $MDP \setminus R$, trajectories \mathcal{T} , trajectories' weights ω_i , $i = 1, \dots, N$, learning rate α

Initialize reward parameter θ randomly

Repeat

Learn the policy π_θ

Compute $L = \sum_i \sum_{(s,a) \in \xi_i} \omega_i \log(\pi_\theta(s, a))$

Update $\theta = \theta + \alpha \nabla L$

Until target number of iterations completed

3.3 Expectation-maximization IRL

To deal with trajectories with multiple reward functions, i.e., multiple strategies, Babes-Vroman et al. [27] proposed a straight-forward expectation-maximization IRL (EM-IRL). Herein, we adapted the original EM-IRL to automatically determine the optimal number of clusters. Instead of directly assigning the cluster number, we considered a possibly maximal number of clusters, i.e., K_{max} , and a variable k initialized as 2 indicating the current cluster number.

Specifically, to determine the optimal number of clusters, starting from the cluster number $k = 2$, we iteratively implemented the EM procedure, until a pre-defined *stop_criteria* was met. The *stop_criteria* was defined as: either there were some empty clusters generated or the log-likelihood (LL) of the clustering results defined in Eq. 5 varied smaller than a pre-defined threshold comparing to the last iteration, which we set as 10. The LL reflected the clustering performance by measuring the accordance of learned clusters with the correspondingly induced cluster-wise strategies. In Eq. 5, N_j stands for the number of trajectories in cluster j .

$$LL = \sum_{j=1}^k \sum_{i=1}^{N_j} \log(z_{ij}) \quad (5)$$

$$z_{ij} = Pr(\xi_i | \theta_j) = \prod_{(s,a) \in \xi_i} \frac{\pi_{\theta_j}(s, a) \rho_j}{Z}, \quad (6)$$

Before the EM loop, parameters ρ_j and θ_j , $j = 1, \dots, k$, which denoted the estimated prior probability and reward parameter for the j^{th} cluster were randomly initialized.

In the **E step**, the probability that trajectory i belongs to cluster j was calculated by Eq.6, in which Z is a normalization factor; In the **M step**, the prior probability of cluster is updated by Eq. 7. Meanwhile, the reward parameter θ_j can be learned by any IRL and herein we employed the MLIRL with weights of trajectories being z_{ij} .

$$\rho_j = \sum_i \frac{z_{ij}}{N} \quad (7)$$

The E step and M step will be iteratively executed until a target number of iterations is completed, which was set as 80 in this work to ensure the convergence. Finally, we found k clusters when LL got converged, with each cluster standing for a group of trajectories with an unique reward function. Based on these reward functions, we could further induce the cluster-wise strategies.

4. SIMULATION ENVIRONMENTS

Algorithm: EM-IRL

Input $MDP \setminus R$, trajectories \mathcal{T} , maximal number of clusters K_{max}

Initialize $k = 2$

While $k \leq K_{max}$

Initialize ρ_j and θ_j , $j = 1, \dots, k$, randomly

Repeat

E Step: Compute the z_{ij} , $i = 1, \dots, N$

M Step: Update the prior probability ρ_j ; and

Learn reward parameter θ_j via MLIRL

Until target number of iterations completed

If *stop_criteria* is True: **Break**; **Else**: $k = k + 1$

Since the ground-truth of students' subtypes were unknown in advance, it is difficult to directly evaluate the EM-IRL learned clusters from the students' data. Thus, we first carried out EM-IRL in three simulation environments which had decided ground-truth. If different strategies could be accurately distinguished by EM-IRL in simulations, we would be more confident to further deploy it in ITS environment.

4.1 Environment Settings

We explored three simulation environments including Grid World, Highway, and Mountain Car, as shown in Figure 1.

Grid World: adapted from [27], in which three grids were randomly chosen as puddles indicated by bricks in Figure 1(a).

- **States (25)** 5×5 grid-size.
- **Actions (4)** Moving to up, down, left, or right.
- **Strategies (3)** Moving to the 1) upper-right corner; 2) lower-left corner; or 3) lower-right corner.

The rewards are designed for the three strategies: 1) Upper-right corner has the reward of 10; 2) Lower-left corner has the reward of 10; 3) Lower-right corner has the reward of 10. Otherwise, each state was punished -1.

Highway: adapted from a three-lane highway scenario introduced in [50], in which the agent controlled a blue car with three speed levels, which could switch between the three lanes or go off-road on either side. At all timestamps, there would be a red car in one of the three lanes.

- **States (729)** the blue car's speed had 3 levels and could move horizontally in 9 locations; the red car could move vertically in 9 locations and horizontally in 3 locations.
- **Actions (5)** Staying at the current state, speeding up, slowing down, moving left, or moving right.
- **Strategies (2)** 1) Keeping off the left lane (suppose it is under construction); 2) Driving at the fastest speed.

The rewards are designed for the two strategies: 1) Driving on the left lane has the reward of -10; 2) Driving with the lowest level of speed has the reward of -10. In both strategies, off-road is punished -0.5, collision is punished -5, and maintaining the state has no reward.

Mountain Car: adapted from the MountainCar-v0 in OpenAI Gym [51], in which a car was on a one-dimensional track and moves between two mountains.

- **States (80)** 10 horizontal positions with 8 levels of speed.

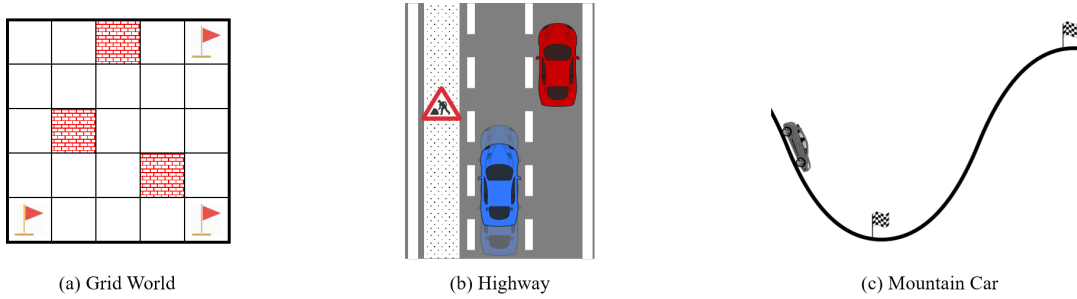


Figure 1: Three simulation environments: (a) Grid World; (b) Highway; (c) Mountain Car.

Table 1: Cluster-wise and overall purities by EM-IRL clustering in three simulation environments.

Environment	Cluster-wise		Overall Purity (%)
	Strategy Idx	Purity (%)	
Grid World	1	100	100
	2	100	
	3	100	
Highway	1	100	100
	2	100	
Mountain Car	1	100	96.4
	2	93.2	

- **Actions (3)** Pushing left, no pushing, or pushing right.
- **Strategies (2)** 1) Reaching to the right mountain top (the car needs to drive back and forth to build up enough momentum to push up); 2) Parking at the valley bottom.

The rewards are generated for the two strategies: 1) Right mountaintop has the reward of +10; 2) Valley bottom has the reward of +10. Otherwise, each state is punished -1.

In each environment, the initial states were randomly assigned, the transitions between states were stochastic and estimated from the data. For each strategy, we induced a policy via policy iteration and employed it to collect trajectories. Specifically, the number of collected trajectories for each strategy was 500, 1000, and 1000 in three environments, respectively. In each environment, trajectories with various strategies were mixed together and fed into the EM-IRL.

Given the ground-truth of cluster-belongings in simulation environments, the results of EM-IRL were evaluated by the purity of each cluster and across overall clusters. Denote the size of i^{th} cluster as N_i with ground-truth labels \mathbf{L}_i , then the cluster-wise purity is calculated as the number of majority labels divided by the cluster size, i.e., $purity_i = \frac{majority(\mathbf{L}_i)}{N_i}$; and the overall purity is calculated by the mean of purity among all clusters, i.e., $purity = \frac{1}{k} \sum_{i=1}^k purity_i$.

4.2 EM-IRL Results in Three Simulations

The EM-IRL clustering results for the three simulation environments are shown in Table 1, in which the first column is the environment; second and third columns show the index

of strategy and the corresponding cluster-wise purity; the last column show the overall purity among all clusters.

In Grid World, all strategies could be accurately clustered by EM-IRL. Specifically, both cluster-wise purities and overall purity were 100%. Likewise, in Highway, the two strategies were accurately clustered with the purity of 100%. In Mountain Car, a few trajectories of driving to right mountaintop (Strategy 0) were mis-clustered to the parking at valley (Strategy 1). This is because the mis-clustered trajectories tried to move to left to collect enough momentum, which showed very similar behaviors to reaching the valley. Overall, the results suggested the effectiveness of EM-IRL in accurately distinguishing subtypes of trajectories with different strategies in all three simulation environments.

5. ITS LEARNING ENVIRONMENT

Our data was collected by letting students work on a web-based ITS, which taught college students probability, e.g., Addition Theorem and Bayes' Theorem. The instruction was conducted by guiding students go through training problems. For each problem, the tutor provided step-by-step instruction, immediate feedback, and on-demand help. The help was provided via a sequence of increasingly specific hints. The last hint in the sequence, i.e., the bottom-out hint, told the student exactly what to do. During training, the students could make pedagogical decisions on whether to solve the next step by themselves or observe the tutor to solve it. If they choose to solve by themselves, the tutor will *elicit* the solution from them by asking questions; otherwise, the tutor will show or *tell* them the solution directly.

5.1 Data Collection

All students participating in our data collection went through four phases: textbook, pre-test, training, and post-test. During *textbook*, all students studied the domain principles from a probability textbook. They read a general description of each principle, reviewed some examples of it, and solved some single- and multiple-principle problems. Then the students took a *pre-test* which contained 14 problems. During this phase, they would not be given feedback on their answers, nor be allowed to go back to earlier questions (this was also true for the post-test). During the *ITS training* procedure, students received 12 problems in the same order. Each main domain principle was applied at least twice. The minimal number of steps needed to solve each training problem ranged from 20 to 50. Such steps included variable definitions, principle applications, and equation solving. The

number of domain principles required to solve each problem ranged from 3 to 11. Finally, all students took the *post-test* which contained 20 problems in total. 14 of the problems were isomorphic to the problems given in the pre-test phase, while the remaining 6 were harder non-isomorphic multiple-principle problems.

The pre- and post-tests required students to derive an answer by writing and solving one or more equations. We used three scoring rubrics: binary, partial credit, and one-point-per-principle. Under the binary rubric, a solution was worth 1 point if it was completely correct or 0 if not. Under the partial credit rubric, each problem score was defined by the proportion of correct principle applications evident in the solution. A student who correctly applied 4 of 5 possible principles would get a score of 0.8. The one-point-per-principle rubric in turn gave a point for each correct principle application. All of the tests were graded in a double-blind manner by a single experienced grader. The results we presented were based upon the partial-credit rubric but the same results hold for the other two. For comparison purposes, all test scores were normalized to the range of [0, 100].

We measure students' learning performance using normalized learning gain (NLG), which measured their gain *irrespective of their incoming competence*. It is calculated as: $NLG = \frac{post - pre}{100 - pre}$, where *pre* and *post* refer to the students' test scores before and after the ITS training respectively and 100 is the maximum score. Herein, for the post-test, we considered all 20 problems that are either isomorphic and non-isomorphic. In addition, an isomorphic NLG (Iso_NLG) was also measured. Unlike NLG, the Iso_NLG was calculated based on the pre- and isomorphic post-test scores, which contained only 14 isomorphic multiple-principle problems.

5.2 States & Actions

Our dataset contains 127 students. Each student spent ~ 2 hours on the system and completed around 400 steps.

States 142 state features were extracted from the student-system interaction log data. Specifically, the features can be grouped into five categories:

- **Autonomy** (10 features): the amount of work done by a student, such as the number of elicits since the last tell;
- **Temporal** (29): time related information about the student's behavior, such as the average time per step;
- **Problem Solving** (35): information about the current problem solving context, such as problem difficulty;
- **Performance** (57): information about the student's performance so far, such as the percentage of correct entries;
- **Hints** (11): information about the student's hint usage, such as the total number of hints requested.

For each category, we employed K-means clustering to get the discretized states. By selecting an elbow of errors when the clustering results got converged, the number of states for each category of features was determined as follows: Autonomy (3 states), Temporal (4), Problem Solving (3), Performance (4), and Hints (3). As a result, we got 432 discrete states totally. Based on the discretized states, we estimated the transition probabilities from all available data.

Actions The students can take two action of elicit/tell, i.e., to *elicit* the solution by themselves through asking questions, or to let the tutor *tell* them the solution directly.

6. EXPERIMENTAL SETTINGS

6.1 Student Subtyping by EM-IRL

Based on the EM-IRL learned clusters, we conducted analyses by checking the statistical significance among different clusters' learning performance, including the pre-test scores, isomorphic NLG (Iso_NLG), NLG, students' learning time on the training task (Time), and the percentage of elicit in students' decisions (Elicit_Perc).

6.2 Student Subtyping by Other Methods

6.2.1 Clustering by Traditional Methods

To evaluate the clustering performance of EM-IRL, we compared it with three other clustering methods: two K-means based approaches that took the pre-test scores and the learning state in the final step as the input respectively and a K-medoids based approach that took dynamic time warping (DTW) [52] distance between trajectories as the input. The K-means based approaches were static-information-based clustering while the K-medoids based DTW considered dynamic state transitions in the trajectories. In our experiments, each of these methods generated three clusters and for each cluster, the MLIRL was employed to learn a strategy. Based on the learned strategies, we calculated the log-likelihood (LL, referring to Eq. 5) of observing such clustering results.

6.2.2 Clustering by Matching RL / IRL Policies

We further explored whether RL or IRL policies could model the heterogeneity in student decision-makings. The inducing of these two policies are detailed as follows.

Inducing the RL policy: To investigate whether students' learning strategies could be distinguished from the tutor's perspective, we compared students' decisions to a RL induced pedagogical policy and clustered the students based on the matching rate. Since the RL policy was induced with the goal of improving students' learning performance, it is expected that the group with a higher matching rate with the RL policy would have better learning performance.

Specifically, we applied RL to learn a pedagogical policy that determines whether the next step should be elicit or tell (the same decisions students made in our ITS). The training data set contained 1,118 students' interaction logs collected from a series of seven prior studies which followed the identical procedure and learning materials as the students in this study described in Section 5. The same 142 features used by EM-IRL were extracted from the logs and used to induce the policy. In an empirical classroom study, the policy was compared with a deep Q-network (DQN) induced policy and a random policy. Results showed that the RL policy significantly outperformed both of them [21].

Once the RL policy was induced, we applied it on the student decision-making data (127 students) to see what decision the RL policy would make on each step. Then, we calculated the matching rate between students' decisions and the RL policy individually for each student. Based on the matching

rates, the students were split into three groups via K-means clustering, denoted as High, Medium, or Low based on the average matching rate of the group.

Inducing the IRL Policy: Similarly, to investigate whether students' learning strategies could be distinguished from their own perspective, we applied IRL to induce a policy from student decision-making data and compared students' decisions with the IRL policy. Given that our data analysis showed that most of students learned significantly from ITS training, herein, we assumed that a majority of students completed the training with the goal to learn. Thus, we expected that the group with a higher matching rate with the the IRL policy would have better learning performance.

The IRL policy was induced from the 127 students who were given the opportunities to make pedagogical decision during training. Herein, the MLIRL algorithm [27] was utilized for policy induction. Similar to the RL based method, the IRL policy was applied back to students' data to calculate the matching rate between students' decisions and the IRL policy. Then, K-means clustering was applied on the matching rate to cluster students into High, Medium, or Low groups.

7. RESULTS

7.1 Student Subtyping by EM-IRL

Fitting students' data to the EM-IRL framework in Section 3.3, when *stop_criteria* was met, we got three clusters. Table 2 shows the EM-IRL subtyping results. From left to right, it shows the students' subtypes, number of students (# Stu), pre-test score (Pre), isomorphic NLG (Iso_NLG), NLG, time on the training task (Time), and percentage of elicit in students' decisions (Elicit_Perc). Based on statistical analysis, we named the three resulting clusters as: *learning-oriented*, *efficient-oriented*, and *no learning*.

A one-way ANOVA analysis on pre-test scores showed no significant difference among the three clusters: $F(2, 124) = 1.36$, $p = 0.260$, $\eta = 0.022$. This suggested that students in the three clusters were balanced in incoming competence. To measure students' learning gain in training, we conducted analyses on their Iso_NLG and NLG. A one-way ANOVA analysis on Iso_NLG showed a significant difference among the three clusters: $F(2, 124) = 3.24$, $p = 0.042$, $\eta = 0.050$. Subsequent contrast analysis revealed that *learning-oriented* > *no learning*: $t(124) = 2.54$, $p = 0.012$, $d = 0.75$ and *efficient-oriented* > *no learning*: $t(124) = 2.19$, $p = 0.030$, $d = 0.54$. Similar results were found for NLG in that a one-way ANOVA analysis showed a significant difference among the three clusters: $F(2, 124) = 3.73$, $p = 0.027$, $\eta = 0.057$. Subsequent contrast analysis revealed that *learning-oriented* and *efficient-oriented* significantly outperformed *no learning*: $t(124) = 2.73$, $p = 0.007$, $d = 0.77$ and $t(124) = 2.15$, $p = 0.033$, $d = 0.52$ respectively.

In terms of time on task, a one-way ANOVA analysis showed a significant difference among the three clusters: $F(2, 124) = 5.81$, $p = 0.004$, $\eta = 0.086$. Subsequent contrast analysis indicated that *learning-oriented* took longer time on task than the other two clusters: $t(124) = -3.11$, $p = 0.002$, $d = 0.58$ for *efficient-oriented* and $t(124) = 2.37$, $p = 0.019$, $d = 0.63$ for *no learning*. A contrast analysis on the percentage of elicit in students' decisions revealed that *learning-*

oriented took significantly more elicit actions than *no learning*: $t(124) = 2.24$, $p = 0.027$, $d = 0.70$.

To summarize, the *learning-oriented* subtype spent significantly more time than the other two groups on the training task and achieved the best performance on both Iso_NLG and NLG (significantly higher than *no learning*). This suggested that learning-oriented students mainly focused on learning the materials, regardless of the time they may spend. The *efficient-oriented* subtype significantly outperformed *no learning* on learning performance and at the same time spent significantly less time than *learning-oriented*. This suggested that *efficient-oriented* students could balance learning gain and time on task. Finally, the *no learning* subtype achieved the lowest learning outcomes.

7.2 Student Subtyping by Other Methods

7.2.1 Clustering by Traditional Methods

We compared our EM-IRL with three traditional baseline clustering methods, namely K-means on the pre-test score (K-means on Pre); K-means on the learning state (142 features) in the final step (K-means on Final Step); K-medoids on the DTW distance among trajectories [52], which is calculated based on the 142 features (K-medoids on DTW). The results are shown in Table 3, with the two columns being clustering method and the resulting log-likelihood (LL).

Overall, results showed that the dynamic-information-based clustering approaches (K-medoids on DTW and EM-IRL) performed better than static-information-based approaches (K-means on Pre and K-means on Final Step). Between the two static-information-based approaches, K-means on final Step performed better than K-means on pre-test. This is not surprising because the state in the final step included information generated during training while the pre-test score only included information till the end of pre-test. Between the two dynamic-information-based approaches, EM-IRL outperformed K-medoids on DTW. A possible reason is that EM-IRL took both states and actions into account while K-medoids on DTW considered only the states in trajectories.

7.2.2 Clustering by Matching RL / IRL Policies

Results of Matching with the RL Policy: Based on the matching rate with the RL policy, we got three clusters by K-means: High ($M = .84$, $SD = .05$), Medium ($M = .70$, $SD = .05$), and Low ($M = .52$, $SD = .07$). A one-way ANOVA analysis over the matching rate showed a significant difference: $F(2, 124) = 339.87$, $p < 0.0001$, $\eta = 0.846$. Subsequent contrast analysis showed that: High > Medium: $t(124) = 4.38$, $p < 0.0001$, $d = 0.99$ and Medium > Low: $t(124) = 8.01$, $p < 0.0001$, $d = 1.70$.

A one-way ANOVA analysis on pre-test showed there was no significant difference among the three groups: $F(2, 124) = 0.26$, $p = 0.771$, $\eta = 0.004$. Analyses on Iso_NLG (calculated based on pre-test and isomorphic post-test) and NLG (calculated based on pre-test and full post-test, which contains six additional hard problems) also showed no significant difference among the three groups. In terms of time on the training task, there was a significant difference among the three groups: High ($M = 2.40$, $SD = .50$), Medium ($M = 2.42$, $SD = .66$), and Low ($M = 1.88$, $SD = .40$).

Table 2: EM-IRL clustering results in ITS environment.

Subtype	#Stu	Pre	Iso_NLG	NLG	Time	Elicit_Perc (%)
learning-oriented	50	73.9(16.8)	55.9(45.3)	23.4(53.6)	2.52(.70)	87.53(13.40)
efficient-oriented	64	76.2(14.5)	43.9(92.4)	-4.4(127.2)	2.18(.45)	84.93(15.02)
no learning	13	81.9(17.4)	-21.1(212.1)	-98.4(340.4)	2.10(.50)	77.06(20.04)

Table 3: Comparison of the log-likelihood (LL) for different clustering methods

Method	LL ($\times 10^3$)
K-means on Pre	-10.68
K-means on Final Step	-9.60
K-medoids on DTW	-8.83
EM-IRL	-6.36

A one-way AVONA on time shows: $F(2, 124) = 9.21$, $p = 0.0002$, $\eta = 0.129$. Subsequent contrast analysis revealed that the High and Medium groups spent significantly more time than the Low group: $t(124) = 3.85$, $p = 0.0002$, $d = 1.11$ and $t(124) = 3.99$, $p = 0.0001$, $d = 0.92$, respectively. An analysis on the percentage of elicit in students' decisions showed a significant difference among the three groups: $F(2, 124) = 66.97$, $p < 0.0001$, $\eta = 0.519$. Subsequent contrast analysis revealed that High > Medium: $t(124) = 4.38$, $p < 0.0001$, $d = 0.99$ and Medium > Low: $t(124) = 8.01$, $p < 0.0001$, $d = 1.70$.

The results showed that by matching with the RL strategy, we could differentiate students' time-consuming strategies from time-efficient strategies. However, it was not able to identify the student subtypes that made a difference in the learning performance. This suggested the presence of a gap between tutor's and students' strategies. Specifically, comparing to taking actions following the tutor's decisions passively, the students might prefer actively direct their own learning process. Therefore, when deploying the tutor's strategy to students, it might not promote the learning performance as expected.

Results of Matching with the IRL Policy: Based on the matching rate with the IRL policy, we got three clusters by K-means: High ($M = .86$, $SD = .05$), Medium ($M = .71$, $SD = .05$), and Low ($M = .54$, $SD = .06$). A one-way ANOVA analysis over the matching rate showed a significant difference among the three groups: $F(2, 124) = 360.99$, $p < 0.0001$, $\eta = 0.853$. Subsequent contrast analysis showed that: High > Medium: $t(124) = 15.92$, $p < 0.0001$, $d = 3.37$ and Medium > Low: $t(124) = 13.52$, $p < 0.0001$, $d = 3.23$.

A one-way ANOVA analysis on pre-test showed there was no significant difference among the three groups: $F(2, 124) = 1.17$, $p = 0.314$, $\eta = 0.019$. Analyses on the Iso_NLG and NLG also showed no significant difference among the three groups. In terms of time on the training task, there was a significant difference among the three groups: High ($M = 2.44$, $SD = .54$), Medium ($M = 2.27$, $SD = .68$), and Low ($M = 2.08$, $SD = .42$). A one-way AVONA on time shows: $F(2, 124) = 3.11$, $p = 0.048$, $\eta = 0.048$. Sub-

sequent contrast analysis showed that the High group spent significantly more time than the Low group: $t(124) = 2.43$, $p = 0.017$, $d = 0.70$. An analysis on the percentage of elicit in students' decisions showed a significant difference among the three groups: $F(2, 124) = 93.92$, $p < 0.0001$, $\eta = 0.602$. Subsequent contrast analysis revealed that High > Medium: $t(124) = 7.95$, $p < 0.0001$, $d = 1.83$ and Medium > Low: $t(124) = 7.08$, $p < 0.0001$, $d = 1.43$.

The results showed that IRL based policy matching was able to cluster the students' strategies different in time. However, it was unable to learn specific subtype of students whose strategy will lead to better learning outcomes. One possible reason that the IRL-based analyses could not identify the learning-performance-impactful strategies is that a single policy was insufficient to effectively generalize the decision-making patterns for the overall students. Different students might follow heterogeneous decision-making strategies.

In summary, the results suggested that EM-IRL could effectively conduct student subtyping reflecting different decision-making strategies. As a contrast, clustering by traditional methods or by matching RL/IRL policies could not find desired student subtypes.

8. CONCLUSIONS

In this paper, we investigated students' subtyping via EM-IRL. By analyzing students' subtyping, we aimed at putting ourselves in the shoes of students to better understand their decision-making. To evaluate the performance of EM-IRL, we first applied it to three simulation environments, where the EM-IRL displayed robust performance to accurately cluster the trajectories with different strategies. Given the accurate clustering results in simulators, we were more confident to further apply EM-IRL to real world longitudinal students' logs collected from an ITS. The results suggested that the EM-IRL could effectively group students with different subtypes, e.g., learning-oriented, efficient-oriented, and no-learning. As a contrast, clustering by traditional methods or by matching RL/IRL policies could not find desired subtypes. The subtyping results showed the potential of providing tutors evidence to give more customized interventions to better assist students' learning. In the future, we will conduct early clustering to detect students' strategies as early as possible. Besides, empirical studies will be carried out to evaluate the effectiveness of subtyping-based interventions to improve the targeted group of students.

Acknowledgements: This research was supported by the NSF Grants: Generalizing Data-Driven Technologies to Improve Individualized STEM Instruction by Intelligent Tutors(2013502); CAREER: Improving Adaptive Decision Making in Interactive Learning Environments(1651909); Integrated Data-driven Technologies for Individualized Instruc-

tion in STEM Learning Environments(1726550); MetaDash: A Teacher Dashboard Informed by Real-Time Multichannel Self-Regulated Learning Data(1660878).

9. REFERENCES

- [1] Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.
- [2] Vincent AWMM Aleven and Kenneth R Koedinger. An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. *Cognitive science*, 26(2):147–179, 2002.
- [3] Arthur C Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M Louwerse. Autotutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers*, 36(2):180–192, 2004.
- [4] Wilson J González-Espada and Daniel W Bullock. Innovative applications of classroom response systems: Investigating students’ item response times in relation to final course grade, gender, general point average, and high school act scores. *Electronic Journal for the Integration of Technology in Education*.
- [5] Ana Iglesias, Paloma Martínez, Ricardo Aler, and Fernando Fernández. Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems. *Knowledge-Based Systems*, 22(4):266–270, 2009.
- [6] Anna N Rafferty, Emma Brunskill, Thomas L Griffiths, and Patrick Shafto. Faster teaching via pomdp planning. *Cognitive science*, pages 1290–1332, 2016.
- [7] Min Chi, Kurt VanLehn, Diane Litman, and Pamela Jordan. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction*, 21(1-2):137–180, 2011.
- [8] Guojing Zhou, Jianxun Wang, Collin Lynch, and Min Chi. Towards closing the loop: Bridging machine-induced pedagogical policies to learning theories. In *EDM*, 2017.
- [9] Roger Azevedo, Nicholas V Mudrick, Michelle Taub, and Amanda E Bradbury. Self-regulation in computer-assisted learning systems. 2019.
- [10] Philip H Winne and Allyson F Hadwin. Studying as self-regulated learning. In *Metacognition in educational theory and practice, The educational psychology series*. 1998.
- [11] Philip H Winne and Allyson F Hadwin. The weave of motivation and self-regulated learning. In *Motivation and self-regulated learning*, pages 309–326. 2012.
- [12] Jeffrey Alan Greene and Roger Azevedo. A theoretical review of winne and hadwin’s model of self-regulated learning: New perspectives and directions. *Review of educational research*, 77(3):334–372, 2007.
- [13] Claire Ellen Weinstein, Jenefer Husman, and Douglas R Dierking. Self-regulation interventions with a focus on learning strategies. In *Handbook of self-regulation*, pages 727–747. Elsevier, 2000.
- [14] Michelle Taub, Nicholas V Mudrick, and Roger Azevedo. Strategies for designing advanced learning technologies to foster self-regulated learning. *Strategies for deep learning with digital technology: Theories and practices in education*, pages 137–170, 2017.
- [15] Min Chi, Kurt VanLehn, Diane Litman, and Pamela Jordan. An evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach. *International Journal of Artificial Intelligence in Education*, 21(1-2):83–113, 2011.
- [16] Shayan Doroudi, Kenneth Holstein, Vincent Aleven, and Emma Brunskill. Towards understanding how to leverage sense-making, induction and refinement, and fluency to improve robust learning. *International Educational Data Mining Society*, 2015.
- [17] Kenneth R Koedinger, Emma Brunskill, Ryan SJD Baker, Elizabeth A McLaughlin, and John Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 34(3):27–41, 2013.
- [18] Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1077–1084, 2014.
- [19] Jonathan P Rowe and James C Lester. Improving student problem solving in narrative-centered learning environments: A modular reinforcement learning framework. In *AIED*, pages 419–428. Springer, 2015.
- [20] Shitian Shen and Min Chi. Aim low: Correlation-based feature selection for model-based reinforcement learning. *International Educational Data Mining Society*, 2016.
- [21] Guojing Zhou, Hamoon Azizzoltani, Markel Sanz Ausin, Tiffany Barnes, and Min Chi. Hierarchical reinforcement learning for pedagogical policy induction. In *International conference on artificial intelligence in education*, pages 544–556. Springer, 2019.
- [22] Guojing Zhou, Xi Yang, Hamoon Azizzoltani, Tiffany Barnes, and Min Chi. Improving student-tutor interaction through data-driven explanation of hierarchical reinforcement induced pedagogical policies. In *Proceedings of the 28th Conference on User Modeling, Adaptation and Personalization*. ACM, 2020.
- [23] Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. 2008.
- [24] Hideki Asoh, Masanori Shiro1 Shotaro Akaho, Toshihiro Kamishima, Koiti Hasida, Eiji Aramaki, and Takahide Kohro. An application of inverse reinforcement learning to medical records of diabetes treatment. In *ECMLPKDD2013 workshop on reinforcement learning with generalized feedback*, 2013.
- [25] Anna N Rafferty, Michelle M LaMar, and Thomas L Griffiths. Inferring learners’ knowledge from their actions. *Cognitive Science*, 39(3):584–618, 2015.
- [26] Anna N Rafferty, Rachel Jansen, and Thomas L Griffiths. Using inverse planning for personalized feedback. *EDM*, 16:472–477, 2016.

- [27] Monica Babes, Vukosi Marivate, Kaushik Subramanian, and Michael L Littman. Apprenticeship learning about multiple intentions. In *Proceedings of the 28th International Conference on Machine Learning*, pages 897–904, 2011.
- [28] Manuel Ignacio Lopez, Jm M Luna, C Romero, and S Ventura. Classification via clustering for predicting final marks based on student participation in forums. *International Educational Data Mining Society*, 2012.
- [29] M Durairaj and C Vijitha. Educational data mining for prediction of student performance using clustering algorithms. *International Journal of Computer Science and Information Technologies*, 5(4):5987–5991, 2014.
- [30] Mohammad Khalil and Martin Ebner. Clustering patterns of engagement in massive open online courses (moocs): the use of learning analytics to reveal student categories. *Journal of Computing in Higher Education*, 29(1):114–132, 2017.
- [31] Cen Li and Jungsoo Yoo. Modeling student online learning using clustering. In *Proceedings of the 44th annual Southeast regional conference*.
- [32] Shitian Shen and Min Chi. Clustering student sequential trajectories using dynamic time warping. *International Educational Data Mining Society*, 2017.
- [33] Saleema Amershi and Cristina Conati. Automatic recognition of learner groups in exploratory learning environments. In *International Conference on ITS*, pages 463–472. Springer, 2006.
- [34] Saleema Amershi and Cristina Conati. Combining unsupervised and supervised classification to build user models for exploratory. *JEDM Journal of Educational Data Mining*, 1(1):18–71, 2009.
- [35] Mirjam Köck and Alexandros Paramythis. Activity sequence modelling and dynamic clustering for personalized e-learning. *User Modeling and User-Adapted Interaction*, 21(1-2):51–97, 2011.
- [36] Vincent Aleven and Kenneth R Koedinger. Limitations of student control: Do students know when they need help? In *International conference on ITS*, pages 292–303. Springer, 2000.
- [37] Ido Roll, Ryan SJ d Baker, Vincent Aleven, and Kenneth R Koedinger. On the benefits of seeking (and avoiding) help in online problem-solving environments. *Journal of the Learning Sciences*, 23(4):537–560, 2014.
- [38] Antonija Mitrovic and Brent Martin. Scaffolding and fading problem selection in sql-tutor. In *Proceedings of the 11th International Conference on Artificial Intelligence in Education*, pages 479–481, 2003.
- [39] Yanjin Long and Vincent Aleven. Mastery-oriented shared student/system control over problem selection in a linear equation tutor. In *International conference on intelligent tutoring systems*, pages 90–100. Springer, 2016.
- [40] Guojing Zhou, Collin Lynch, Thomas W Price, Tiffany Barnes, and Min Chi. The impact of granularity on the effectiveness of students’ pedagogical decisions. In *CogSci*, pages 2801–2806, 2016.
- [41] Guojing Zhou and Min Chi. The impact of decision agency & granularity on aptitude treatment interaction in tutoring. In *CogSci*, pages 3652–3657, 2017.
- [42] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.
- [43] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.
- [44] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.
- [45] Andrew Y Ng and Stuart J Russell. Algorithms for inverse reinforcement learning. In *ICML*, volume 1, page 2, 2000.
- [46] Shao Zhifei and Er Meng Joo. A survey of inverse reinforcement learning techniques. *International Journal of Intelligent Computing and Cybernetics*, 5(3):293–311, 2012.
- [47] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pages 2586–2591, 2007.
- [48] Christos Dimitrakakis and Constantin A Rothkopf. Bayesian multitask inverse reinforcement learning. In *European workshop on reinforcement learning*, pages 273–284. Springer, 2011.
- [49] Jaedeug Choi and Kee-Eung Kim. Nonparametric bayesian inverse reinforcement learning for multiple reward functions. In *Advances in Neural Information Processing Systems*, pages 305–313, 2012.
- [50] Edouard Klein, Matthieu Geist, Bilal Piot, and Olivier Pietquin. Inverse reinforcement learning through structured classification. In *Advances in NIPS*, pages 1007–1015, 2012.
- [51] Praveen Palanisamy. *Hands-On Intelligent Agents with OpenAI Gym: Your guide to developing AI agents using deep reinforcement learning*. Packt Publishing Ltd, 2018.
- [52] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

Analyzing Student Procrastination in MOOCs: A Multivariate Hawkes Approach

Mengfan Yao
Department of Computer
Science
University at Albany - SUNY
myao@albany.edu

Shaghayegh Sahebi
Department of Computer
Science
University at Albany - SUNY
ssahebi@albany.edu

Reza Feyzi Behnagh
Department of Educational
Theory & Practice
University at Albany - SUNY
rfeyzibehnagh@albany.edu

ABSTRACT

Student procrastination, as the voluntary delay of intended work despite expecting to be worse off for the delay, is an important factor with potentially negative consequences in student well-being and learning. In online educational settings such as Massive Open Online Courses (MOOCs), the effect of procrastination is considered to be even more prevalent and detrimental, as online courses are often self-paced and self-directed, where higher levels of self-regulated learning are expected from the students. Past research has mainly described students' procrastination by either static time-related measures (e.g. averaged starting time over all assignments per student), or by temporal models' parameters, under the assumptions that student activities take place at a constant rate (e.g. Homogeneous Poisson models), and that student interactions with one learning material are independent of interactions with another. In this work, we propose to consider the interdependence between the students' temporal activities while modeling their sequences in a continuous time scale. To this end, we propose to model the interaction sequence between each student and each course module, i.e. each module-student pair, as Multi-dimensional Hawkes processes, which not only capture the relationship between students' learning activities and their exogenous stimuli such as assignment deadlines, but also capture the endogenous responses *within* and *between* types of learning materials. Our experiments show that not only there exists dependencies between students' historical activities and the future ones when different types of learning materials are involved, such dependencies also provide meaningful interpretations in terms of students' procrastination behaviors. Furthermore, our findings show that in addition to association with delay, the parameters learned by multi-dimensional Hawkes processes provide more procrastination-related information and can improve our explanation of student grades.

Keywords

Procrastination, MOOCs, Student Modeling, Multivariate Hawkes Process, Clickstream Data

Mengfan Yao, Shaghayegh Sahebi and Reza Feyzi Behnagh "Analyzing Student Procrastination in MOOCs: A Multivariate Hawkes Approach" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 280 - 291

1. INTRODUCTION

Student academic procrastination has shown to have negative effects on students' learning and well-being. Procrastination is prevalent in different academic settings like traditional classrooms, and could be even more widespread in online learning environments, as higher levels of time-management and self-regulated learning (SRL) skills are required [47, 3, 16]. To describe and measure student procrastination, past research has been mainly relying on either self-reported surveys (e.g. [27]) or time-related features that are associated with students' dilatory behaviors (e.g. [10]). As procrastination is inherently subjective, self-reported surveys have been heavily used in earlier research, to differentiate procrastinators and non-procrastinators by emphasizing on measuring the perceptions of the students. Although self-report survey measures capture students' retrospective reports of their studying and delaying behaviors, they are administered in a cross-sectional manner, rely on students' memory, and are usually static point estimates that summarize students' *average* degree of procrastination.

Considering the noises of self-reported data [37], in more recent studies, more focus has been given to the behavioral side of the procrastination, where time-related measures were proposed and used as the representation of students' procrastination. For example, measures such as students' average delays in starting coursework, the average time they spent in doing assignments, students' average paces of viewing lectures have been studied as factors of procrastination [2, 10, 15, 6, 22]. However, these measures lack the ability to describe students' continuous behaviors within a period of time. An analogy to such methods is to describe the entire distribution using the sample mean, without fully knowing the distribution. To tackle this limitation, more recently, emphasis has been on modeling the time points of student activities that are extracted from students' learning trajectory data (e.g. log or click-streams of student historical actions), via stochastic models. For example in [33], Park et al. modeled students' per-day activity counts during each week of the course via a Poisson mixture model, which models the entire trajectory of each student activities during a weekly module. Other factors that have been considered to be important in describing procrastination in the past research are the effects of different learning materials (e.g. forums and quizzes) as well as students' interactions with them (e.g. [1, 28]). However, to the best of our knowledge, no past work has considered the possible time dependencies within and between students' interactions with dif-

ferent learning material types. For example, viewing video lectures more intensively *mostly before* the first attempt of an assignment may suggest that a student prefer to learn the materials first before trying the assignment. On the other hand, watching lecture videos *dominantly after* the first attempt of an assignment may suggest that the student prefer to try the assignment first and then go through the video lectures if they encountered any problems.

To summarize, past research has attempted to describe procrastination using static time measures, or measures summarized from more sophisticated temporal models, based on students' interactions with one or more learning materials. However, two important factors of student behaviors and their association with procrastination have not been fully explored: (1) the dependencies between students' past and future interactions within each learning material type (e.g. knowing a student has looked at lecture slides at some time, how and when are they going to have the next activity?) and (2) the dependencies between students' interactions with different types of learning materials (e.g. are watching video lecture usually followed by a submission of an assignment?) In this work, we aim to address these two factors by answering the following questions: within each learning module, that is the unit of a course that learning materials are provided, (Q1) are the past activities independent of future ones? Or some activities can trigger other ones to arrive within a short period of time (i.e. time dependencies between activities)? And (Q2), are students' interactions with one type of learning material (e.g., video lectures) independent from another type (e.g., discussion forums)? Furthermore, (Q3) if such dependencies exist, how are they associated with student procrastination? (i.e. the dependencies between a student's past and future activities as well as dependencies a student's interactions with one learning material with another.)

As a result, our goal is to find the missing link between students' procrastination and students' activities within and between different types of online learning materials. To achieve this goal, we propose to use multi dimensional Hawkes processes as a powerful tool that addresses the above mentioned concerns in student procrastination analysis. Particularly, we represent all activities on one type of learning material as one dimension in the multi-dimensional Hawkes model. We show that this model better fits our data, in comparison to baseline temporal processes. Also, to answer Q1 and Q2, we demonstrate that it can capture both students' reactions to the deadlines as action-triggering factors that come externally (i.e. exogenous stimuli), and students' responses to the previous interactions with different types of learning materials, such as video lectures, assignments, and discussions (i.e. self-excitement). By doing so, we can understand students' procrastination behavior from a stochastic process point of view, with two main stimuli: (1) some of the students' activities can be viewed as a response to an external stimulus, e.g. deadlines of the assignments (2) some other student activities can be viewed as the results of previous interactions that the student had with the same or other learning material types. Based on the model parameters, to answer Q3, we also propose a measure that not only describes student procrastination but also is able to explain student performance better than the static delay measure.

The outline of this paper can be summarized as follows: In Section 2, we go over three main bodies of the related work; in Section 4 we go over the details of the dataset that we use; in Section 4, we provide the intuition of using the Hawkes model, then statistically and visually show that a Hawkes process is a proper choice for modeling module-student interactions; in Section 5, we formally define our problem and introduce the multi-dimensional Hawkes model that we use in this study. We perform various experiments in section 6, to analyze the model parameters, explain their interpretation, and associate them with procrastination as well as students' assignment grades. Finally, the conclusion of this work is summarized in Section 7.

2. RELATED WORK

Students' procrastination In the past research on student procrastination, the main focus has been on the measures that capture either students' perceptions (e.g. self-reported surveys on procrastination [35, 41, 40, 12, 23]), or static measures that describe students' dilatory behaviors as the representation of procrastination [15, 11, 44, 33]. For example, in [10], Cerezo et al. studied 140 undergraduate and used measures such as students' delay and time-spent variables to describe procrastination. For another example, in [2], Asarta and Schmidt studied students' behaviors in accessing lecture notes of a blended-learning course, and proposed to use features such as pacing, anti-cramming, and consistency in reviewing course materials. A few recent works have tried to model student activities to provide a temporal perspective of procrastination behavior. For example, Backhage et al. proposed a model that captures procrastination-deadline cycles of all students in the course using a stochastic temporal model [4]. However, this model assumes that all students follow the same procrastination behavior during the course and does not distinguish the differences between student behaviors. In [33], Park et al. assumed that students' daily activity counts follow a mixture Poisson distribution, which is a mixture of a procrastination component and a non-procrastination component. Particularly, by assuming the independence between students' past and future activities, they proposed to model each day of the week by a Poisson with a constant rate for all weeks. In the end, they described procrastinators as the ones with a dominant procrastination component versus the non-procrastination one, i.e. the students who have a fast-increasing activity counts towards the end of the week. Moon et al. assumed that procrastination behavior over time can be described by a curvilinear growth curve and modeled it using latent growth curve modeling. To validate this assumption, they compared the curvilinear model with a non-growth and a linear model and showed that their model has a better goodness-of-fit than the baseline models [30]. In contrast with the existing research that either uses summary variables or ignores the dependence between different student activities, in this paper we aim to model the temporal activity interrelationships and associate them with student procrastination.

Modeling students' engagement using their learning trajectories. Other relevant studies to our work are the ones that model student learning trajectories to understand other aspects of their behaviors, such as student engagement in online learning environments. While many past

studies focused mostly on utilizing cumulative factors such as frequency of watching videos or using discussion forums [39, 17, 34, 13], more recent work attempted to build more complex models of student behaviors. For example, in [46], Zhu et al. constructed students' social connection networks based on students' weekly post-reply dynamics, along with node attributes, such as assignment scores. Particularly, they used an exponential random graph model to compute the structural features of the social connection networks, to understand the relationship between students' engagement in the forums and their performances in the assignments. In another example, Lan et al. proposed a statistical model, which consists of two components: a learning model and a response model [25]. These two models represent nine behavioral features extracted from students' video-watching clickstreams and in-video quiz responses in one MOOC course, with the aim to find the behavioral features that lead to high levels of student engagement. Similarly, Kizilcec et al. classified students' behaviors based on binary features extracted from students' log data (1 if a student had any activities that are associated with a learning material, 0 otherwise, for all learning materials) [24]. As a result, they identified 4 behavioral types: completing, auditing, disengaging, and sampling, from these binary features. For another similar example, Gelman et al. extracted student features from students' log data as well and applied Nonnegative Matrix Factorization to find 5 types of student behaviors - deep, consistent, bursty, introduction, and sampling [18]. Particularly, the authors used a procrastination indicator as a feature, that is, the average amount of time left before the deadline when a student submits their assignments. In summary, past research on modeling student engagement is similar to our study in the sense that the models utilize students' activities that were extracted from students' log data. However, it differs to ours in the following two ways: (1) the models usually define students' engagement levels based on the *counts* of students' historical activities, without directly modeling students' learning trajectories as stochastic processes, (2) the aim is usually to model or predict students' future engagement levels, rather than studying students' procrastination and its association with students' performance.

Hawkes in education. Hawkes processes, a family of stochastic point processes, have been frequently used to model complicated time-stamped events in continuous time. Due to Hawkes process's capability to model scenarios where historical events influence future activities, it has been frequently used in finance [5] and seismology [32] and has been gradually becoming a useful modeling tool in the domain of social media [7, 36, 29], as well as recommendation systems [14, 43, 21, 38]. In the education domain, a few works have used Hawkes processes so far, especially to model social and interaction data among students [19, 20, 26]. For example, Lan et al. proposed a single-dimensional Hawkes model to recommend relevant discussion threads to students according to their historical interactions with course forums. In a similar application, a Hawkes model is suggested by Von Davier et al. to model the collaboration dynamics between students within and between groups [42]. Along this line, Halpin et al. used multi-dimensional Hawkes processes to understand students' collaboration with each other [19, 20]. Another interesting application of the Hawkes process in the

education domain is the work by Boerner et al. that analyzed the association between student skills and the skills required by professional jobs [8]. In another recent work, Cai et al. used Hawkes processes as a step in their model to predict which video a student will watch next based on their historical interactions with the videos in an edX course [9]. They use long and short multi-dimensional Hawkes processes that differentiates the long-term and short-term temporal dependencies between video-watching actions. None of the above works uses the Hawkes processes to model the procrastination behavior, nor considers course deadlines and milestones in their application of the Hawkes process.

3. DATASET

Our dataset is publicly collected from the Canvas Network¹ MOOC platform [31], which is an online platform that hosts various open online courses in different academic disciplines, such as Computer science, Social Science, and Business management. These courses have multiple types of learning resources, including Wiki pages, assignments (or quizzes), and discussions. Assignments can be quiz-style or in a longer format where students need to upload a file to complete the submission. Each learning module is associated with one Wiki page. In total, CANVAS data contains 389 anonymized courses where the names of students and courses along with the contents of discussions and assignment (or quiz) submissions are not available.

In this work, we mainly focus on exploring the student learning trace data. Specifically, we select a computer science course (course id: 770000832960058) that best fits the following criteria: (1) having a large number of students²; (2) including multiple types of learning materials (such as video lectures, assignments, discussions); and (3) containing a large number of student historical learning activities. To obtain student learning activity data, we use Canvas logs files (Pageview requests). We divide the learning activities into three types. Specifically, we consider viewing the lectures, downloading the attached files, and previewing the attached files as the activities associated with video lectures (*L*). Activities that include viewing, creating, saving, updating, and submitting each assignment attempt are associated with assignments (*A*). Finally, we consider reading (marking as read), subscribing, creating, replying, and editing discussion entries, discussion topics, and direct messages as discussion-related activities (*D*).

We separate the data in module-student pairs, as we aim to model each student's interactions with each individual learning module. As each module has its specific deadlines according to the course design, we choose each module rather than the whole course as the unit of our study. Also, different modules usually have different learning objectives, which will possibly trigger different behaviors. By doing so, we are able to capture a finer granularity of the data. Finally, we have 731 students and $\sim 946K$ learning activities in the selected course.

¹<http://canvas.net>

²Enrolled students who have missed more than 50% of the assignment submissions during the courses, along with those who did not receive a final grade, are considered as dropouts and are disregarded in this study.

4. BACKGROUND: HAWKES PROCESS AS A FIT TO STUDENT ACTIVITIES

Since we want to study the interactions between students and modules from a temporal aspect, point processes are one of the best choices for our application. Additionally, because of the interaction irregularities in our application, we must select a point process that can handle this type of information. Specifically, past studies have shown that students' activities can take place in an irregular manner during various periods of a course, particularly affected by milestones such as assignment deadlines and exam dates [16]. As a result, the point processes that follow a constant rate, such as Poisson processes, are not the appropriate model for our application. In Poisson processes, the main assumption is the independence between past and future occurrences of the events, which can not be met in student studying behaviors. Not only some student activities happen in response to the course milestones, but also a part of these activities can be interrelated with each other. For example, a student whose goal is to start discussing a topic in the discussion forum may watch a video lecture about the same topic before posting in the forum. To meet the temporality and interdependence assumptions of our application, we choose to model student activities during course modules with Hawkes processes.

One of the most important properties of the Hawkes process is its ability to deal with the interrelationships between future and past activities. This is in contrast with the memoryless Poisson process where all activities are assumed to be independent of each other. More importantly, the Hawkes process allows the activities to be excited both exogenously (by external stimuli, similar to the Poisson process) and endogenously (self-excitement, by internal stimuli). In other words, the Hawkes process has a branching process point of view. It assumes that some activities arrive as a result of exogenous stimuli (i.e. immigrant activities). Then, the immigrant activities can trigger their following activities (i.e. offspring activities), and those offspring activities can further trigger their own offspring activities, and so on. That is, the offsprings of an immigrant activity are structured into a latent cluster because they are all triggered by the same immigrant and arrive more closely to each other than the activities that are in other clusters.

As a result, the Hawkes process can capture more information than the Poisson process or other point processes that use the average base rate as the only model parameter. This can be very helpful when modeling processes that have the same number of activities, but with different activity occurrence distributions. To demonstrate this ability in Hawkes processes, we show the event occurrence patterns of two simulated Hawkes processes with the same number of activities, but different parameters in Figure 1. We can see that process 1 has more bursty but less regular occurrences compared to process 2, in which less burstiness but a higher regularity is observed. Since both simulated Hawkes processes have the same number of activities in the history, a Poisson model is not able to capture such differences because the base rates of the two processes would be the same in a Poisson process.

For an educational application, there is a natural mapping between student activity events and Hawkes processes. The smaller student activity chunks toward a goal or deadline can

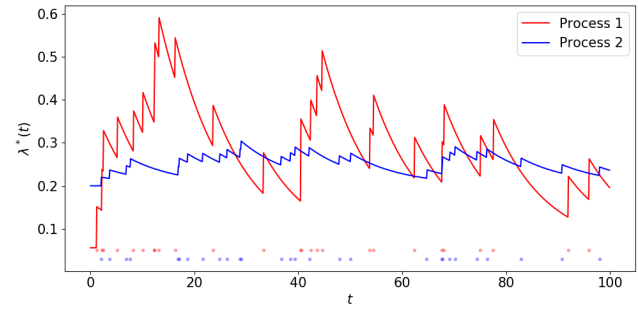


Figure 1: An illustration of two different processes that have the same number of occurrences. The x-axis is the time and the y-axis is the intensity of event occurrences per time unit. Both processes have 29 occurrences but with very different characteristics that will be ignored by Poisson processes.

be examples of an immigrant's offsprings: students break down the big tasks (the whole process) into small sub-tasks (latent clusters). The deadline (external stimuli) of a big task, such as an assignment deadline, can trigger subsequent activities that are associated with the small tasks. These activities arrive closely one after another in a so-called bursty manner (self-excitement)³. We demonstrate that Hawkes processes are a good fit to our application by showcasing two examples. First, we show that the module-student pair interactions can not be properly modeled by processes that only model an average base rate, such as Poisson processes. To do this, we conduct a goodness of fit test on the inter-arrival times of module-student pairs in our dataset against the inter-arrival time distribution of a Poisson process, which is $\exp(1)$. We use the Kolmogorov-Smirnov test to evaluate the fit's significance. The mean p -value of this test among all module-student pairs in our dataset is $2.77E-6$ with a standard deviation of $6.41E-5$, which shows that module-student pairs do not fit Poisson processes.

Second, we empirically demonstrate the burstiness of module-student interactions. To do this, we show that the Poissonian property of only having a constant base rate is not present in the observed activities of a sample module-student pair from our dataset. Specifically, we use the 1-lag autocorrelation of activity inter-arrival times to conduct our test. The inter-arrival time is defined as the difference between the arrival times of two consecutive activity occurrences. We first simulate a Poisson process with the base rate equal to the average number of activities in our sample activity sequence. Then, we compare the 1-lag autocorrelation in this simulated sequence with the autocorrelation of our sample sequence. Since all inter-arrival times in Poisson processes follow $\exp(1)$, we expect the autocorrelation of the simulated Poisson process to be 0 (no correlation). In contrast, we expect to see a non-zero autocorrelation in

³It is worth noting that in regular applications of Hawkes processes an activity at time t can trigger later activities at times $\tau > t$. However, in our application, student activities are triggered by the upcoming deadlines in the future. Similarly, earlier chunks of studying sub-tasks at times $\tau < t$ can be offsprings of future studying tasks at time t towards a deadline. As a result, to make the Hawkes process applicable to our problem, we use a reversed activity timeline for our data. This does not affect our model, optimization, or learned parameters.

a bursty self-exciting sequence. Figure. 2 shows the scatter plot of activity inter-arrival times in the original sequence vs. the sequence with lag 1 for each of the two sequences. As we can see, little autocorrelation is spotted in the Poisson process, whereas the pattern of autocorrelation in real data is shown to be not random. Specifically, we can see that most of the lag-1 vs. original inter-arrival times for the sample sequence are scattered around the axes, meaning that dense activities are often followed by long pauses, and vice versa.

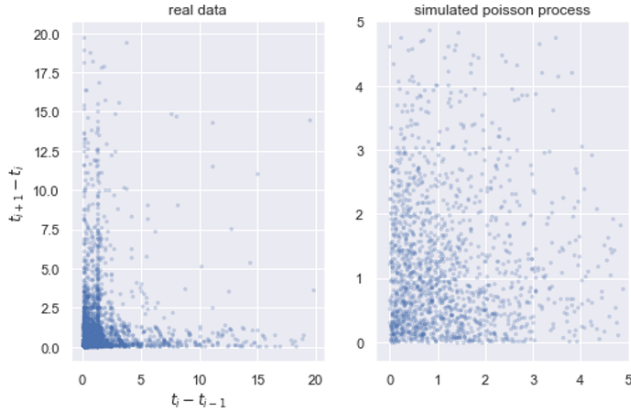


Figure 2: A demonstration of burstiness presented in the interactions of a sample module-student pair: 1-lag autocorrelation scatter plots shows that long pauses are often observed after dense and bursty activities.

It is worth mentioning that our goal is not to directly compare to Poisson models. Rather, we are demonstrating here that we must model the data in a way that captures long-term temporal properties of the processes and their irregularities, rather than static measures such as the count or average number of activities that only provide one facet of the whole picture.

5. METHOD: MULTI-DIMENSIONAL HAWKES PROCESSES TO MODEL ACTIVITY-TYPE RELATIONS

In this section, we introduce the method we use in this study to model student behaviors. More specifically, we illustrate multi-dimensional Hawkes processes and how we apply them to our application. The previous section illustrated how Hawkes process is a good fit for student activities as future activities in module-student pairs could be related to the past activities. In those illustrations, all activities in a module-student pair are considered to be homogeneous or of one single type. In other words, the self-exciting property of the interactions between students and module are assumed to be uniform throughout different kinds of activities, whether it is watching a video lecture, participating in a discussion, or attempting to submit a solution to an assignment. However, in reality, students might exhibit different learning behaviors or use different learning strategies towards different types of learning materials. For example, some students may have more intense and frequent activities when viewing module lectures but less frequent pace when it comes to the discussions. Furthermore, when a student is interacting with two different types of learning materials, different time dependencies may exist between student's in-

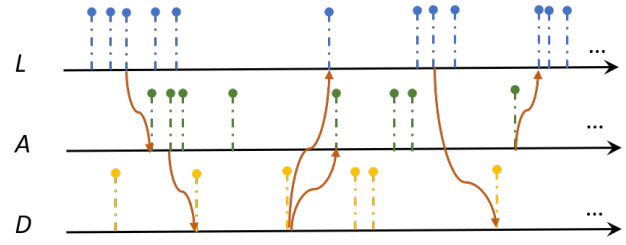


Figure 3: Hawkes processes in module-lecture dimension L , module-assignment dimension A , and module-discussion D , and their mutual excitation. A vertical bar is the representation of an activity occurrence and a red arrow shows the influence of one activity (head) on another (end).

teractions with the two. For example, a student may often visit discussion forums very closely after viewing lecture slides, i.e. strong time dependency between lectures and discussions (more specifically, discussion after lectures), but such dependencies may be less obvious for another student.

To address this challenge, we model the students' activities on one type of learning material as an individual Hawkes process, and model all such processes simultaneously as multi-dimensional Hawkes Processes. In particular, in the rest of this study, we refer the collection of activities that associate with one type of learning material as a Hawkes process *dimension*. A multi-dimensional Hawkes model not only allows dependency between past and future activities within each dimension (i.e. self-excitation) to be modeled, it is also able to capture the possible dependencies between different types of activities (i.e. excitation between dimensions). For example, scenarios such as submitting the first attempt of an assignment and then starting the second attempt (self-excitation), or, posting a question in the discussion forum after watching a video lecture (excitation between dimensions) can be well described by multi-dimensional Hawkes processes.

In this study, based on the learning material types presented in our dataset, we consider 3 dimensions to analyze students' learning behaviors, namely video lecture dimension L , assignments dimension A and discussions dimension D . To illustrate how multi-dimensional Hawkes processes work in modeling between-dimension excitation, in Figure 3, we show 3 sample Hawkes processes that respectively comes from dimensions L , A , and D . Within each dimension, we use vertical dashed lines to represent the occurrences of activities that take place in that dimension⁴. Activities in one dimension can trigger other activities in another dimensions. This constitutes the influence between different dimensions. We indicate the between-dimension triggers by the red arrows that point from the parent activity to the offspring. For example, the third activity in dimension D in this figure triggers the fifth activity in dimension A as well as the sixth activity in dimension L .

We now formally explain the multi-dimensional Hawkes model and how it can be interpreted according to our application. Suppose that for each module-student pair (m, u) , we are

⁴The height of each bar does not represent intensity and does not have any particular meaning in this figure

given a sequence of arrival times for N_{mu} number of activities that are associated with module m and student u . We represent the sequence of each module student pair as in $(m, u) = \{\tau_i\}_{i=1}^{N_{mu}}$, where $\tau_i = (t_i, d_i)$ corresponds to the arrival time of i^{th} activity and the dimension (activity type) d_i to which activity i belongs. For example, suppose student u has 3 total activities in module m . If u submitted an assignment at time 1, then checked a lecture's slides at time 5, and had some discussion posted at time 8, then $(m, u) = \{(t_1 = 1, d_1 = A), (t_2 = 5, d_2 = L), (t_3 = 8, d_3 = D)\}$, with A , L and D representing assignments, video lectures, and discussions respectively. For each dimension $d \in [L, A, D]$ and each module-student pair (m, u) , we further use the sequence $T_d(\tau_i) = \{t_i \in \tau_i | \tau_i \in (m, u), d_i = d\}$, to represent the type d learning activities that student u performs in module m as a process. According to the multi-dimensional Hawkes model, we can explain the intensity of $T_d(\tau_j)$ according to the following function:

$$\lambda_d(t) = \mu_d + \sum_{d', t_j < t} \phi_{dd'}(t - t_j), \quad (1)$$

where μ_d describes the average number of activities occurred per unit time that are triggered by exogenous stimuli (the process's base rate in dimension d); and ϕ (the kernel function) represents the function that explains the endogenous stimuli, or the triggering effects from the previous ($t_j < t$) activities in the same dimension or another dimension (d'). In other words, $\phi_{dd'}$ controls the total influence that dimension d exerts on dimension d' , as a function of activity inter-arrival times ($t - t_j$). Using an exponential kernel function for ϕ , the multi-dimensional Hawkes model can be rewritten as in Equation 2.

$$\lambda_d(t) = \mu_d + \sum_{d'} \alpha_{dd'} \sum_{t_j < t} \beta \exp(-\beta(t - t_j)). \quad (2)$$

The term $\alpha_{dd'}$ and the term $\beta \exp(-\beta(t - t_j))$ can be considered as the decomposition of kernel function $\phi_{dd'}$, which respectively describe the influence weight of dimension d on dimension d' (including α_{dd} , the self-excitation of dimension d itself) and an exponential decay function $g(t) = \beta \exp(-\beta(t))$. Putting together all activity types' parameters, we use a d -dimensional vector $\mu = [\mu_d]$ to represent the base rates of the processes in all dimensions, and a $d \times d$ matrix $\Phi = [\phi_{dd'}]$ to represent the between and within dimension triggering effects. From here, we can write $\Phi = I \circ G$, where we have *influence matrix* $I = [\alpha_{dd'}]$ and *exponential decay kernel* $G = [\sum_{t_j < t} (-\beta \exp(-\beta(t - t_j)))]$. Based on that, we can also describe the aggregated influence of dimension d on other dimensions using the following equation:

$$\alpha_d = \frac{1}{|\{d\}|} \sum_{d'} \alpha_{dd'}, \quad (3)$$

which is simply the average influence of dimension d over all dimensions. A summary of all notations used so far are shown in Table 1.

This intensity function $\lambda_d(t)$ has an intuitive meaning: all the future activities in dimension d , apart from those that are triggered by external stimuli, can be triggered by the previous activities that belong to each of the dimensions d' (including d itself) according to the influence weight $\alpha_{dd'}$ (the outer summation). The ones that are triggered by ex-

Notation	Description	Formula
L	Dimension module lectures	
A	Dimension module assignments	
D	Dimension module discussions	
(t_j, d_i)	activity j in dimension d_i	
τ_i	arrival time	(t_i, d_i)
(m, u)	module m , student u pair	$\{t_i\}$
$T_d(\tau_i)$	activities in dimension d	$\{t_i \in \tau_i d_i = d\}$
μ_d	base rate in dimension d	
$\alpha_{dd'}$	influence of d to d'	
α_d	Influence of dimension d	$\frac{1}{ \{d\} } \sum_{d'} \alpha_{dd'}$
β	decay parameter	
$g(t)$	decay kernel function	$\beta \exp(-\beta(t))$
$\phi_{dd'}(t)$	Hawkes kernel function	$\alpha_{dd'} g(t)$
I	influence matrix	$[\alpha_{dd'}]$
G	decay kernel matrix	$[\sum_{t_j} (-\beta \exp(-\beta(t - t_j)))]$
Φ	Hawkes kernel matrix	$I \circ G$
λ_d	intensity in d	Equation 1

Table 1: Notations and their descriptions.

ternal stimuli take places with rate μ_d . Furthermore, as a past activity becomes distant (larger $t - t_j$), its effect on the occurrence probability of a new event decreases exponentially (i.e. the inner summation). From the branching process point of view, the kernel function $\phi_{dd'}$ is designed in this way so that $\alpha_{dd'}$ is the branching ratio. By computing $\frac{1}{1 - \alpha_{dd'}}$, we can obtain the expected number of future activities in dimension d' that are triggered by an immigrant in dimension d . This represents the size of an offspring cluster.

To avoid possible confusions, we also want to clarify that by saying one activity i in dimension d' *triggers* another activity j in dimension d , we mean that the probability of activity j in the result of activity i is higher than j coming from base rate μ_d or triggered by other activities. To see this, one can interpret Equation 1 as follows: in dimension d , a sequence of activities come from the base rate μ_d and each summation leads to a sequence of activities with parameter $\phi_{dd'}$. Then, the probability of j being triggered by i is

$$P(j \text{ child of } i) = \frac{\phi_{dd'}(t_j - t_i)}{\mu_d + \sum_{t_i < t_j} \phi_{dd'}(t_j - t_i)}. \quad (4)$$

Parameter Estimation. A common way to find the best parameters of Hawkes model, given the observed activity arrival times, is to minimize the negative log-likelihood of the data. Particularly, given the sequence $\{(t_i, d_i), \dots, (t_N, d_N)\}$ till some time T , the log-likelihood of having influence matrix I and base rate vector μ is of the following form:

$$\mathcal{L}(I, \mu) = \sum_{i=1}^N \log(\mu_d + \sum_{t_j < t} \alpha_{dd'} g(t_i - t_j)) - T \sum_{d=1}^n \mu_d - \sum_d \sum_{d'} \alpha_{dd'} \int_0^{T-t_j} \phi(T - t_j) dt_j. \quad (5)$$

In order to find Hawkes parameters that models each module-student pair, we adopted algorithm ADM4 [45], which made use of a mix of Lasso and nuclear regularization on top of the negative log-likelihood. Specifically, Accelerated Projected Gradient Descend method was used to meet the non-negative constraints on I and μ as Hawkes parameters only have realistic meanings when the parameters are non-negative⁵. When it comes to the selection of global parameter β , for

⁵We made our implementation available at <https://github.com/ssahebi/EDM2020-Hawkes>.

each module-student pair, we use grid search with cross validation on the interval $[0, 10]$ with step size 1.

6. EXPERIMENTS

6.1 Testing the Goodness of Fit

To test the goodness-of-fit of the model, in Table 2, we compare the RMSE of the intensity for all dimensions, computed based on the observed module-assignment pairs, for multi-dimensional Hawkes model (i.e. Equation 1), single-dimensional Hawkes model and a Poisson model. Specifically, for single-dimensional Hawkes, we treat all activities as in one dimension, and estimate the intensity of each dimension using the uni-variate parameters α , β , and μ . For the Poisson model, in each dimension, we use the average activity arrival rate as the base rate, and compute the RMSE for each dimension respectively. As we can see in

	L	A	D
Hawkes (Multi)	0.56	2.34	1.37
Hawkes (Single)	0.71	2.57	1.95
Poisson	3.22	6.91	3.73

Table 2: The goodness of fit to true data for each model in terms of intensity RMSE.

Table 2, Poisson has the worst fit in all dimensions, possibly caused by the non-Poissonian nature we showed from the real data. Single-dimensional Hawkes has comparable but slightly worse performance. One possible reason is that there might exist differences in terms of base rate and burstiness between dimensions and by modeling all types of learning materials as one activity type, the model can only capture the average trend in all dimensions. To visualize how the multi-dimensional Hawkes processes fit the real data, we also present in Figure 4 the estimated intensity (blue) and true intensity (black) of a sample module-student pair. As we can see in this figure, the model mostly has a good fit to the real data. Only at some time points, it underestimates the expected number of activities that are about to happen.

6.2 Model Parameter Analysis: Trends and Differences Between Dimensions

In this section, we analyze the estimated Hawkes parameters within and between different dimensions, to show their general trends and differences across different dimensions.

We start this part with a correlation analysis of all Hawkes Parameters, to show the general trends and possible differences between dimensions. Particularly, we calculate the Spearman rank correlation coefficients between the parameters that are learned for all module-student pairs as is shown in Figure 5. Recall that parameters $\alpha_{dd'}$, μ_d , β and α_d respectively is the between-dimension (or within if $d = d'$ excitation, base rate, decay rate (Equation 2) and aggregated influence of dimension d (Equation 3) for $d \in [L, A, D]$.

We can see that self-excitation within dimensions (i.e. α_{dd}) are generally negatively correlated with base rates μ_d of the same dimensions and decays β . This means that as the external stimuli leads to more and more expected arrivals, i.e. when regular activities come from the base rate, the effect of each previous activity on the future ones tends to decrease, i.e. self-excitation gets weaker. In other words, in

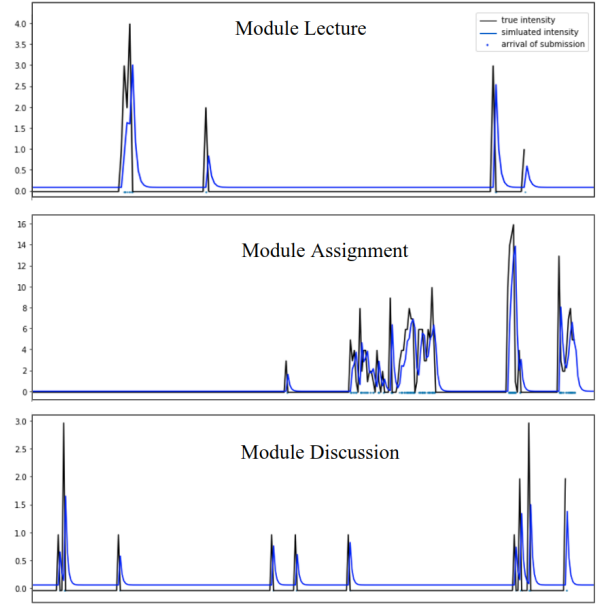


Figure 4: Estimated and true intensity of a sample module-student pair, modeled by multi-dimensional Hawkes.

sequences with higher regularity, less burstiness is observed. Also, it means that activities that have a slower decay rate usually arrive in a more bursty manner. Mapping to our application of students interacting with learning modules, by using the branching process point of view, higher α suggests higher expected number of activities in a latent cluster as sub tasks. On the other hand, the negative correlation also means a lower base rate and lower number of immigrants. In other words, the number of such latent clusters are also fewer. One possible interpretation is that in each dimension, students divide their big learning task into sub tasks and work for each individual sub task in a relatively bursty manner. This also suggests that students barely have behaviors that are both highly intense and highly frequent (i.e. large μ_d and α_{dd}). Similarly, both highly sparse and highly mediated activities (i.e. small μ_d and α_{dd}) are rarely observed neither, as the correlation between μ_d and α_{dd} is positive for all $d \in [L, A, D]$.

Comparing the parameter correlations across different activity types, we can see that dimensions L and D , have high within and between-dimension influence correlations. For example, the correlation between α_{LL} and α_{LA} is 0.97 in dimension L and correlation between α_{DL} and α_{DA} is 0.96. This implies that the influence of discussion and video lecture activities on other dimensions are almost consistent. For example, if the influence of video lectures on assignments is high, it is likely that the influence of video lectures on discussions is high too. Similarly, if the pattern of interacting with video lectures in a module is bursty (high α_{LL}), it is likely that other activity types triggered by video lectures are also bursty. However, the influence of assignment activities on discussions and video lectures are not significantly associated with assignment activity's self-excitement. That could mean that after a student has a bursty set of assignment activities in a module, the student is less likely to have

a bursty video lecture activities. Similarly, the influence of assignment activities on discussions has a low correlation with the influence of assignment activities on video lectures. For instance, if a student starts an assignment intensively very closely after some intensive watching of video lectures, the student is less likely to have high intense discussion activities after assignments. We can also see that assignment-triggered activities' burstinesses are less correlated with the base rates (i.e. α_{AD} vs. μ_d). This means that the frequency of activities that come from external stimuli does not affect the influence of assignment activities on consequent activities in other dimensions. Taken altogether, it is interesting to see that activities that are associated with assignments tend to have different exciting patterns compared to video lectures and discussions. This can show the influence of deadlines, as the assignments are the only activity type that have deadlines and are going to reflect student grades in this dataset.

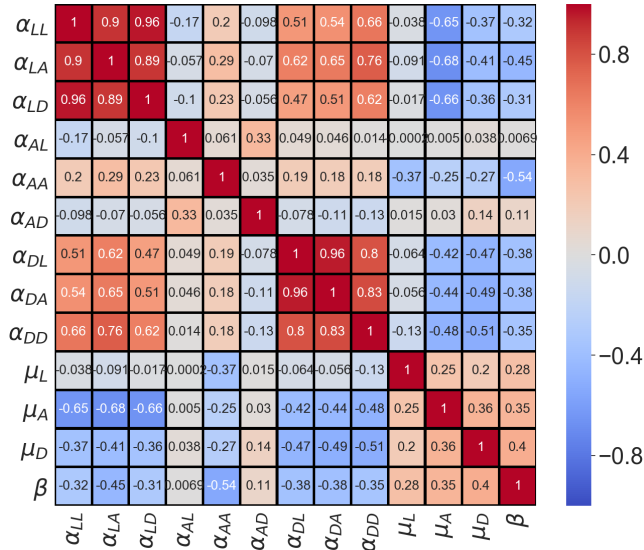


Figure 5: Spearman Rank Correlation Coefficients between Hawkes Parameters.

6.3 Student Behaviors Characterized by Model Parameters

In the previous part, we were interested in showing the correlation between Hawkes parameters that represented within and the between-dimension relationships. In this part of the analysis, we focus on the different behaviors that are observed according to the learned parameters for each module-student pair. Additionally, we are interested to see if these learned Hawkes parameters are proper representatives for student procrastination. To do so, we first define a measure that can represent student procrastination in the absence of self-reported data. In the following, we go over some important assumptions, definitions and time measures that we use for procrastination.

Defining Delay as a Procrastination Measure. We assume that each student works on one module at a time, meaning that they do not work on several modules at the same time. Furthermore, we assume that submitting the last attempt of the module's assignment marks the end of study-

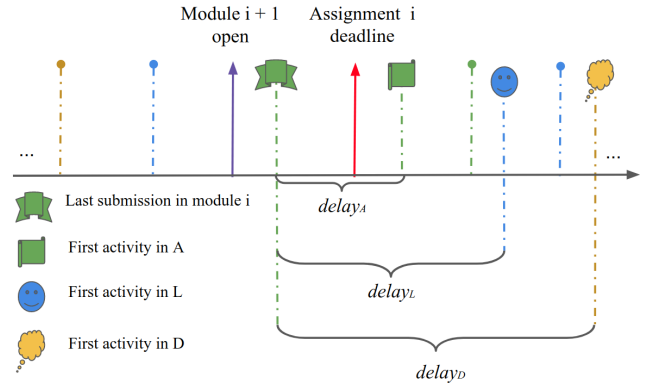


Figure 6: Illustration of delay measures. As in Figure 3, we use blue, green and yellow dashed lines to represents activities from dimension L, A and D respectively.

ing this module. According to these assumptions, we define module i 's end time for student j , t_{ij}^e as the time stamp when the last module assignment was submitted. We then define the start time t_{ij}^s as the earlier time stamp between $t_{i-1,j}^e$ and the available time for module i . In other words, when student j finishes learning module $i-1$, if module i has already been made available, then their end time on module $i-1$ is defined as the start time for module i . Otherwise, the start time is going to be the time when module i becomes available or is published online. In each dimension d , we use t_{ij}^d to denote action time, which is defined as the time that the first activity in dimension d takes place between start time t_{ij}^s and end time t_{ij}^e .

Having the module start time and student action time in dimension d , we can calculate how late a student started working on activity of type d in the module using $t_{ij}^d - t_{ij}^s$. To factor in the duration differences between different modules, we normalize this value by the module duration. Eventually, we define the following measure to quantify student j 's normalized delay in dimension d that is associated with module i :

$$delay_d = \frac{t_{ij}^d - t_{ij}^s}{t_{ij}^e - t_{ij}^s} \quad (6)$$

One of the motivations to define the delay according to start time t_{ij}^s is that, sometimes module $i+1$ is available before the assignment deadline in module i . By this time, student j might still be working on module i . So, it would be unfair to count the time after module $i+1$ is available and before student j 's assignment submission as the procrastinating time for student j on starting module $i+1$. On the other hand, if student j finishes the assignment in module i earlier than the deadline of the assignment, this extra time they earned from the early submission can be used toward the next available module. If the student does not use this time, it will be considered as a cramming behavior toward the next module $i+1$. An illustration of these definitions is presented in Figure 6.

Observing Two Behavior Groups. We now focus on the distribution of the learned Hawkes parameters to see if we can observe any behavioral patterns across different student-module pairs. Specifically, in Figure 7 we present

the distribution of the learned α_{LL} , α_{AA} , and α_{DD} . We can clearly observe two spikes in the density distribution of influence parameters, more prominently in α_{LL} and α_{AA} . Combining this observation with the correlation analysis in previous section, we can see that there are two types of module-student interactions: the ones with higher frequency and lower burstiness versus the ones with lower frequency and higher burstiness. To statistically show the dif-

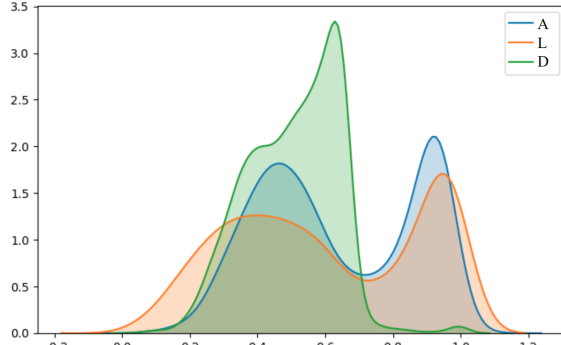


Figure 7: Density distribution of α_M , α_A , α_D .

ference between these two types of interactions, we first cluster student-modules according to their α_{LL} and α_{AA} , into two groups using the K-means clustering algorithm. Then, we test to see if the learned Hawkes parameters, i.e. excitation parameters $\alpha_{dd'}$, base rate μ_d , decay β and aggregated influence α_d for $d \in [L, A, D]$, are statistically different across the two groups. Particularly, we conduct the Kruskal-Wallis test on each learned parameter between the two clusters. The average values of the parameters for each of the two clusters are shown in Table 3. Since the p-values for all tests are smaller than 0.0001, we do not show them in the table. These small p-values suggest that the differences between clusters are statistically significant for all parameters between the two types. This indicate that the differences between the two types are meaningful.

Examining both groups more closely, we can see that, the aggregated influence of dimension A, i.e. α_A , is the highest among all 3 for both type 1 and type 2 groups. With that being said, this influence majorly comes from the self-excitement in dimension A, i.e. α_{AA} . μ_A is also the highest among all 3 dimensions. Combining these observations, we can see that assignment-related activities arrives more frequently and are highly influential in triggering consequent activities, especially the assignment-related ones. Also, we can see that on average type 2 group has a much smaller base rate for video lectures (μ_L) and discussions (μ_D), meaning less density and regularity in those activities compared to type 1. However, In assignment-related activities, the base rate (μ_A) as well as aggregated influence in dimension A (α_A), are higher in type 2 group, which suggests an overall denser and more intense assignment-related activities arrivals comparing to type 1 group.

Now if we look at the differences between two groups in terms of between-dimension relationships, we can see that α_{AL} , i.e. the triggering effect of assignment to consequent video lecture activities (and similarly, α_{AD} : the assignment-triggered discussion activities) is much lower in type 1 group compared to type 2 group. This difference is also notable

in other between-dimension α s. For example, the influence of assignments on video lectures (α_{AL}) is way less than the influence of video lectures on discussions (α_{LD}) in type 2 group, while this difference is less in the type 1 group. This suggests that the interaction patterns with assignments in type 2 group are almost inconsistent with other dimensions. We note that, although the type 1 and type 2 clusters are created according to α_{LL} and α_{AA} parameters only, we see significant differences in all other parameters of the two groups.

Delay in the Discovered Groups. Here, we aim to understand if the two behavior types that we discovered in the previous part are associated with measures of procrastination. Particularly, we evaluate the differences observed in the delay measures defined in Equation 6 for the two clusters. The results are presented in Table 4. Again, all p-values are smaller than 0.0001. A major observation is that type 1 and type 2 have very different delays in all dimensions. Specifically, the delay of each dimension in type 1 group is much less than the corresponding delays in type 2 group. As a result, we can call type 2 group as the delay group and type 1 group as the non-delay group. Given that the type 1 and type 2 behavioral clusters are formed based on Hawkes model parameters only, this important observation demonstrates that the learned Hawkes parameters can clearly represent delay as a procrastination measure. Also, we can see that in delay (type 2) group, on average, students start the first discussion way after the first assignment activity takes place ($delay_D > delay_A$). However, in the non-delay (type 1) group, on average the first assignment activity happens after some discussion ($delay_D < delay_A$). We can see that in both groups, the video lecture activities come before discussions or assignments.

Combined with our observations from the previous analysis, we see that not only the delay group start the first activity in each dimension much later than the other group, they also have a much less base rate μ_L and μ_D . Consequently, we can see that the delay group (type 2) has less frequent but more bursty discussion and lecture-related activities, while the non-delay group activities arrive in a less bursty but more frequent manner in these two dimensions. On the other hand, assignment activities are denser and more intense for the delay group. This combined observation shows that the Hawkes parameters can represent more information about student procrastination, compared to the delay measure alone.

6.4 Student Grades Associated with Model Parameters

In the previous section, we concluded that the learned Hawkes model parameters not only represent delays, but also can capture more procrastination-related behaviors. In the rest of this section, we are interested in exploring if the additional trends captured by the Hawkes model can be more meaningful in association with student grades, compared to the delay parameter. In particular, we are interested in the association between delay and student grades from the Hawkes processes point of view.

Recall that $delay_d$ defined in Equation 6 measures the normalized delay of the first activity in dimension d of the

	d	α_{dL}	α_{dA}	α_{dD}	α_d	μ_d	β
Type 1	L	0.558±0.149	0.263±0.178	0.289±0.185	0.381±0.524	0.0003±0.0006	0.663±0.692
	A	0.107±0.272	0.820±0.125	0.101±0.264	0.462±0.459	0.0003±0.002	
	D	0.322±0.393	0.305±0.393	0.790±0.151	0.394±0.325	5.52E-5±1.8E-4	
Type 2	L	0.874±0.108	0.823±0.135	0.816±0.134	0.795±0.582	4.82E-5±9.86E-5	0.425±0.249
	A	0.019±0.124	0.864±0.061	0.018±0.124	0.799±0.582	0.0004±0.004	
	D	0.699±0.429	0.696±0.430	0.936±0.092	0.590±0.238	1.19E-5±5.92E-5	

Table 3: Statistics of Hawkes parameters $\alpha_{dd'}$, μ_d , β and α_d for $d \in [L, A, D]$ in the two clusters.

	$delay_L$	$delay_A$	$delay_D$
Type 1	0.08±0.228	0.575±0.411	0.338±0.385
Type 2	0.108±0.274	0.722±0.360	0.819±0.337

Table 4: Statistics of delay measures $delay_L$, $delay_A$ and $delay_D$ in two clusters identified by Hawkes parameters.

student-module pair. Here, we define a new delay measure based on both learned parameters of the Hawkes process and $delay_d$. We then study if this newly defined delay measure performs better in association with student grades, compared to $delay_d$. Specifically, after showing the between-dimension excitation interrelationships, it is reasonable to assume that these interrelationships are important in the activity delays as well. For example, knowing that assignment-related activities can trigger followup activities in all 3 dimensions, delaying the assignment-related activities also potentially causes consequent delays in other dimensions. Motivated by this, we propose $delay_d^H$ by combining $delay_d$ and between-dimension Hawkes parameters as follows:

$$delay_d^H = delay_d + \frac{1}{\sum_{d' \neq d} \frac{1}{1-\alpha_{dd'}}} \quad (7)$$

As we mentioned in Section 5, $\frac{1}{1-\alpha_{dd'}}$ can be seen as the statistically expected number of activities in a latent cluster that are triggered by an immigrant. The second term in Equation 7 basically quantifies the potential loss per time unit in terms of triggering other dimensions' activities by delaying in dimension d . Taken altogether, $delay_d^H$ describes the total delays in all 3 dimension that are associated with delay in dimension d .

To see if $delay_d^H$ provides more grade-related information compared to $delay_d$, we look at the Spearman's correlation between these two measures and students' assignment grades. The result of this correlation analysis is presented in Table 5. Our first observation is that the correlations between both delay measures with student grade are negative. However, this correlation is not as significant for $delay_d$, compared to $delay_d^H$. This is specially stronger in the assignment dimension. The reason for this can be two-fold: (1) comparing to $delay_d$, $delay_d^H$ not only captures how late the action was taken in each dimension, it also provides some insights on the student behavior trends throughout their learning process, and (2) as $delay_d^H$ describes the time-dependencies between dimensions, it is more powerful in explaining student activities in all dimensions as a whole, compared to the point estimate summaries of procrastination. Particularly, one may overlook the importance of delaying the discussion-related activities on assignment grades when considering the $delay_D$ measure only. However, a stronger correlation between $delay_D^H$ and grades suggests that early start of the discussion-related activities is almost equally important as starting the video lectures early, probably because of the triggering effect of dimension D and the

d	L	A	D	avg.
$delay_d^H$	-0.339***	-0.125*	-0.329***	-0.264**
$delay_d$	-0.240**	-0.070.	-0.114*	-0.141*

Table 5: Spearman's correlation with respect to assignment score for each delay measure. Significance level is denoted as follows: $p < 0.001$ * $p < 0.01$ ** $p < 0.05$ * $p < 0.1$.**

potential loss that its delay causes to all 3 dimensions.

7. CONCLUSION

In this work, we proposed to use the multi-dimensional Hawkes processes to model procrastination in student learning behavior. We showed that multi-dimensional Hawkes processes have a better fit to student activity counts in comparison with their single-dimensional version and the Poisson processes. By analyzing the correlations between the learned parameters in the Hawkes processes, we concluded that more bursty student sequences have less regular activities in them, the burstiness of video lecture and discussion-related activities vary similar to each other, and the deadlines highly affect the arrival times of assignment-related activities. We showed that Hawkes parameters can reveal two types of behaviors in the data that are associated with different delays - the delay group tends to have high within and between-dimension excitation but low base rate, and the non-delay group have a high base rate and a lower excitation in all dimensions. According to the branching processes point of view, we gave a realistic interpretation on these types of behaviors: non-delay group divide big tasks into many sub-tasks (high base rate) which leads to more frequent and less dense activities throughout the learning process. On the other hand, delay group tend to intensively work in one dimension for a shorter period of time, followed by long pauses (high excitation but low base rate). We also showed that the Hawkes model parameters represent richer information compared to the delay measure alone by defining a new Hawkes-based delay measure and associating it with student grades. Our experiments demonstrated that the between-dimension dependencies in the multi-dimensional Hawkes model better explain student grades.

This study is limited in the number and variety of the datasets that we have experimented on. In the future, we plan to explore more datasets from various disciplines and platforms. Another limitation is the single measure that we use to evaluate procrastination ($delay_d$). As a followup to this study, we aim to define and use more procrastination indicators, including the self-reported procrastination measures.

8. ACKNOWLEDGEMENT

This paper is based upon work supported by the National Science Foundation under Grant Number 1917949.

9. REFERENCES

- [1] J. W. Alstete and N. J. Beutell. Performance indicators in online distance learning courses: a study of management education. *Quality Assurance in Education*, pages 6–14, 2004.
- [2] C. J. Asarta and J. R. Schmidt. Access patterns of online materials in a blended course. *Decision Sciences Journal of Innovative Education*, 11(1):107–123, 2013.
- [3] R. Azevedo and R. Feyzi-Behnagh. Dysregulated learning with advanced learning technologies. volume 7, pages 9–18. Italian e-Learning Association, May 2011.
- [4] C. Backhage, C. Ojeda, and R. Sifa. Circadian cycles and work under pressure: A stochastic process model for e-learning population dynamics. In *Data Science-Analytics and Applications*, pages 13–18. Springer, 2017.
- [5] E. Bacry, I. Mastromatteo, and J.-F. Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005, 2015.
- [6] R. Baker, B. Evans, and T. Dee. A Randomized Experiment Testing the Efficacy of a Scheduling Nudge in a Massive Open Online Course (MOOC). *AERA Open*, 2(4):233285841667400, Oct. 2016.
- [7] P. Bao, H.-W. Shen, X. Jin, and X.-Q. Cheng. Modeling and predicting popularity dynamics of microblogs using self-excited hawkes processes. In *Proceedings of the 24th International Conference on World Wide Web*, pages 9–10. ACM, 2015.
- [8] K. Börner, O. Scrivner, M. Gallant, S. Ma, X. Liu, K. Chewing, L. Wu, and J. A. Evans. Skill discrepancies between research, education, and jobs reveal the critical need to supply soft skills for the data economy. *Proceedings of the National Academy of Sciences*, 115(50):12630–12637, 2018.
- [9] R. Cai, X. Bai, Z. Wang, Y. Shi, P. Sondhi, and H. Wang. Modeling sequential online interactive behaviors with temporal point process. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 873–882, 2018.
- [10] R. Cerezo, M. Esteban, M. Sánchez-Santillán, and J. C. Núñez. Procrastinating Behavior in Computer-Based Learning Environments to Predict Performance: A Case Study in Moodle. *Frontiers in Psychology*, 8, Aug. 2017.
- [11] R. Cerezo, M. Sánchez-Santillán, M. P. Paule-Ruiz, and J. C. Núñez. Students’ LMS interaction patterns and their relationship with achievement: A case study in higher education. *Computers & Education*, 96:42–54, May 2016.
- [12] J. N. Choi and S. V. Moran. Why not procrastinate? development and validation of a new active procrastination scale. *The Journal of social psychology*, 149(2):195–212, 2009.
- [13] R. W. Crues, N. Bosch, M. Perry, L. Angrave, N. Shaik, and S. Bhat. Refocusing the lens on engagement in moocs. In *Proceedings of the fifth annual ACM conference on learning at scale*, pages 1–10, 2018.
- [14] N. Du, Y. Wang, N. He, J. Sun, and L. Song. Time-sensitive recommendation from recurrent user activities. In *Advances in Neural Information Processing Systems*, pages 3492–3500, 2015.
- [15] T. Dvorak and M. Jia. Do the Timeliness, Regularity, and Intensity of Online Work Habits Predict Academic Performance? *Journal of Learning Analytics*, 3(3):318–330, 2016.
- [16] G. C. Elvers, D. J. Polzella, and K. Graetz. Procrastination in online courses: Performance and attitudinal differences. *Teaching of Psychology*, 30(2):159–162, 2003.
- [17] B. J. Evans, R. B. Baker, and T. S. Dee. Persistence patterns in massive open online courses (moocs). *The Journal of Higher Education*, 87(2):206–242, 2016.
- [18] B. Gelman, M. Revell, C. Domeniconi, A. Johri, and K. Veeramachaneni. Acting the same differently: A cross-course comparison of user behavior in moocs. *International Educational Data Mining Society*, 2016.
- [19] P. F. Halpin and P. De Boeck. Modelling dyadic interaction with hawkes processes. *Psychometrika*, 78(4):793–814, 2013.
- [20] P. F. Halpin, A. A. von Davier, J. Hao, and L. Liu. Measuring student engagement during collaboration. *Journal of Educational Measurement*, 54(1):70–84, 2017.
- [21] H. Jing and A. J. Smola. Neural survival recommender. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 515–524, 2017.
- [22] A. M. Kazerouni, S. H. Edwards, T. S. Hall, and C. A. Shaffer. DevEventTracker: Tracking Development Events to Assess Incremental Development and Procrastination. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE '17*, pages 104–109, Bologna, Italy, 2017. ACM Press.
- [23] K. R. Kim and E. H. Seo. The relationship between procrastination and academic performance: A meta-analysis. *Personality and Individual Differences*, 82:26–33, 2015.
- [24] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the third international conference on learning analytics and knowledge*, pages 170–179, 2013.
- [25] A. S. Lan, C. G. Brinton, T.-Y. Yang, and M. Chiang. Behavior-based latent variable model for learner engagement. *International Educational Data Mining Society*, pages 64–71, 2017.
- [26] A. S. Lan, J. C. Spencer, Z. Chen, C. G. Brinton, and M. Chiang. Personalized thread recommendation for mooc discussion forums. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 725–740. Springer, 2018.
- [27] C. H. Lay. At last, my research article on procrastination. *Journal of research in personality*, 20(4):474–495, 1986.
- [28] N. Michinov, S. Brunot, O. Le Bohec, J. Juhel, and M. Delaval. Procrastination, participation, and performance in online learning environments. *Computers & Education*, 56(1):243–252, 2011.
- [29] S. Mishra, M.-A. Rizoïu, and L. Xie. Feature driven and point process approaches for popularity

- prediction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1069–1078. ACM, 2016.
- [30] S. M. Moon and A. J. Illingworth. Exploring the dynamic nature of procrastination: A latent growth curve analysis of academic procrastination. *Personality and Individual Differences*, 38(2):297–309, 2005.
- [31] C. Network. Canvas network courses, activities, and users (4/2014-9/2015) restricted dataset. *Harvard Dataverse*, 1, 2016.
- [32] Y. Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical association*, 83(401):9–27, 1988.
- [33] J. Park, R. Yu, F. Rodriguez, R. Baker, P. Smyth, and M. Warschauer. Understanding student procrastination via mixture models. *International Educational Data Mining Society*, 2018.
- [34] L. W. Perna, A. Ruby, R. F. Boruch, N. Wang, J. Scull, S. Ahmad, and C. Evans. Moving through moods: Understanding the progression of users in massive open online courses. *Educational Researcher*, 43(9):421–432, 2014.
- [35] T. A. Pychyl, J. M. Lee, R. Thibodeau, and A. Blunt. Five days of emotion: An experience sampling study of undergraduate student procrastination. *Journal of social Behavior and personality*, 15(5):239, 2000.
- [36] M.-A. Rizoïu, L. Xie, S. Sanner, M. Cebrian, H. Yu, and P. Van Hentenryck. Expecting to be hip: Hawkes intensity processes for social media popularity. In *Proceedings of the 26th International Conference on World Wide Web*, pages 735–744. International World Wide Web Conferences Steering Committee, 2017.
- [37] A. Rotenstein, H. Z. Davis, and L. Tatum. Early birds versus just-in-timers: the effect of procrastination on academic performance of accounting students. *Journal of Accounting Education*, 27(4):223–232, 2009.
- [38] J. Shang and M. Sun. Geometric hawkes processes with graph convolutional recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4878–4885, 2019.
- [39] K. Sharma, P. Jermann, and P. Dillenbourg. Identifying styles and paths toward success in moocs. *International Educational Data Mining Society*, 2015.
- [40] P. Steel. The nature of procrastination: A meta-analytic and theoretical review of quintessential self-regulatory failure. *Psychological bulletin*, 133(1):65, 2007.
- [41] B. W. Tuckman. Relations of academic procrastination, rationalizations, and performance in a web course with deadlines. *Psychological reports*, 96(3_suppl):1015–1021, 2005.
- [42] A. A. von Davier and P. F. Halpin. Collaborative problem solving and the assessment of cognitive skills: Psychometric considerations. *ETS Research Report Series*, 2013(2):i–36, 2013.
- [43] W. Xiao, X. Xu, K. Liang, J. Mao, and J. Wang. Job recommendation with hawkes process: an effective solution for recsys challenge 2016. In *Proceedings of the recommender systems challenge*, pages 1–4, 2016.
- [44] J. Yoo and J. Kim. Can Online Discussion Participation Predict Group Project Performance? Investigating the Roles of Linguistic Features and Participation Patterns. *International Journal of Artificial Intelligence in Education*, 24(1):8–32, Jan. 2014.
- [45] K. Zhou, H. Zha, and L. Song. Learning triggering kernels for multi-dimensional hawkes processes. In *International Conference on Machine Learning*, pages 1301–1309, 2013.
- [46] M. Zhu, Y. Bergner, Y. Zhang, R. Baker, Y. Wang, and L. Paquette. Longitudinal engagement, performance, and social connectivity: a mooc case study using exponential random graph models. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 223–230, 2016.
- [47] B. J. Zimmerman. Investigating self-regulation and motivation: Historical background, methodological developments, and future prospects. *American educational research journal*, 45(1):166–183, 2008.

Towards Accurate and Fair Prediction of College Success: Evaluating Different Sources of Student Data

Renzhe Yu
University of California, Irvine
Irvine, CA, USA
renzhey@uci.edu

Qiujie Li
New York University
New York, NY, USA
ql16@nyu.edu

Christian Fischer
University of Tübingen
Tübingen, Germany
christian.fischer@uni-tuebingen.de

Shayan Doroudi
University of California, Irvine
Irvine, CA, USA
doroudis@uci.edu

Di Xu
University of California, Irvine
Irvine, CA, USA
dix3@uci.edu

ABSTRACT

In higher education, predictive analytics can provide actionable insights to diverse stakeholders such as administrators, instructors, and students. Separate feature sets are typically used for different prediction tasks, e.g., student activity logs for predicting in-course performance and registrar data for predicting long-term college success. However, little is known about the overall utility of different data sources across prediction tasks and the fairness of their predictions with respect to different subpopulations. Using data from over 2,000 college students at a large public university, we examined the utility of institutional data, learning management system (LMS) data, and survey data for accurately and fairly predicting short-term and long-term student success. We found that institutional data and LMS data both have decent predictive power, but survey data shows very little predictive utility. Combining institutional data with LMS data leads to even higher accuracy than using either alone. In terms of fairness, using institutional data consistently underestimates historically disadvantaged student subpopulations more than their peers, whereas LMS data tend to overestimate some of these groups more often. Combining the two data sources does not fully neutralize the biases and still leads to high rates of underestimation among disadvantaged groups. Moreover, algorithmic biases affect not only demographic minorities but also students with acquired disadvantages. These analyses serve to inform more cost-effective and equitable use of student data for predictive analytics applications in higher education.

Keywords

Predictive analytics; Machine Learning; Higher education; Fairness; Student data

1. INTRODUCTION

The most common application of learning analytics in higher education is using predictive modeling to understand critical factors contributing to student success, or to identify students who need support in a timely manner. Predictive analytics have been used within a course [2] or while using tutoring software [38]. They have also been used to optimize student success in the longer term, for example to predict graduation rates [3] or to make course recommendations [26]. Different data sources can be used to build these predictive models, with varying trade-offs. For example, when making predictions at the course level, log data from learning management systems (LMS) are often used. These systems allow for automated and scalable recording of hundreds of learner actions in every single minute, but they require robust and efficient data management systems. When making longer-term predictions, on the other hand, institutions can use data typically stored in student information systems (SIS), including prior academic history, standardized test scores, and demographic information. While this data source might be readily available to college administrators, it might be more difficult to access, due to ethical concerns or logistic barriers, for individual instructors or researchers trying to build such models for particular use cases. In some cases, both data sources are further combined with assessments or surveys that measure students' metacognitive abilities or other non-cognitive attributes that might predict college success [35]. However, collecting and managing these data is often costly for institutions if they are not already doing so. Given all these trade-offs, it is necessary to examine the utility of different student data sources for building predictive analytics-based solutions to guide instructors, administrators and education policy makers on the costs and benefits of utilizing different data sources.

To date, research that systematically compares data sources and predictions is underrepresented in the literature [14]. To respond to this call for research, this study evaluates the usefulness of three common student data sources for two representative prediction tasks. These three data sources, including institutional data, LMS data, and survey data, are all widely used across research settings and have been shown to predict various measures of college success. Given the different use cases of short-term and long-term predic-

Renzhe Yu, Qiujie Li, Christian Fischer, Shayan Doroudi and Di Xu "Towards Accurate and Fair Prediction of College Success: Evaluating Different Sources of Student Data" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 292 - 301

tions as discussed above, we construct two success measures: individual course grades (short-term success) and yearly average GPA (long-term success). The usefulness of each data source is determined by its contribution to overall prediction accuracy and to prediction fairness across student subpopulations. The focus on fairness arises from the concern that predictive models trained on the entire student population may perform systematically worse on selected subpopulations than other others, which may have unintended negative effects for vulnerable students [6]. For instance, if models are less confident in identifying struggling students among an already underrepresented group, this bias may eventually amplify existing achievement gaps.

In short, our research aims to identify what combinations of student data (a) more accurately predict different success measures; and (b) more fairly predict these measures. The remainder of this paper is organized as follows: Section 2 summarizes the related work on college success prediction and fairness of predictive models; Section 3 describes the data and methods we use to construct and evaluate prediction models; Section 4 presents the results from various predictions; Section 5 reflects on the findings and discusses the practical implications for stakeholders; Section 6 concludes the study with limitations and future work.

2. RELATED WORK

2.1 Predicting College Success Using Student Characteristics

Although college is a complicated ecosystem with numerous factors shaping student outcomes, prior research has identified several groups of student characteristics across institutional data, LMS data, and survey data that consistently predict commonly used measures of success.

2.1.1 Personal Background - Institutional Data

Student success in higher education is often stratified by students' demographic, socioeconomic and academic background prior to college experience. For example, college graduation rates substantially differ by students' race/ethnicity. National data indicates that Hispanic students are 15% less likely to graduate college within six years than their white counterparts, and this gap is 25% between black and white students [33]. Such inequalities are particularly pronounced in STEM fields, where even more underrepresented students drop out of their college careers [1]. Also, student performance prior to entering college (e.g., on standardized tests) has often been found to strongly predict college performance across different subpopulations [9]. These overall trends suggest that what happens before college remains predictive of student success in higher education settings. Of course, this could be due to a variety of factors, such student background being correlated with patterns of historical and institutionalized oppression as well as other barriers that students from different backgrounds might face both before and during college.

2.1.2 Learning Behavior - LMS Data

In contrast to latent psychological states, learning behavior is a more extrinsic and observable predictor of academic success [7]. Behavioral patterns capture variations in college experience that may be orthogonal to students' incom-

ing characteristics, allowing for insights into the mechanism of academic success at a day-to-day granularity. With the prevalence of digital learning platforms, learning behavior can be authentically recorded in the form of clickstream data. These time-stamped data record learner's interactions with LMSs. This allows researchers to create measures that look into the "black box" of study behaviors [5]. For example, how students allocate their study time is a consistent predictor of performance. Those who have more regular engagement patterns and who space out their study effort (instead of cramming) are more likely to be high-achieving [27]. Similarly, students who strategically regulate their learning effort (e.g., starting from exercise-oriented tactics and moving to other tactics based on encountered challenges) perform equally well but with less effort, compared to simply hard-working students [23].

2.1.3 Non-Cognitive Abilities - Survey Data

There is emerging evidence that non-cognitive factors, such as personality traits, task values and self-efficacy, are associated with positive academic outcomes even after controlling for cognitive factors measured by intelligence tests as well as various background characteristics [8]. Among these factors, researchers seem to have reached consensus that self-regulated learning skills are essential because unlike in K-12 schooling, college students have the flexibility as well as responsibility to actively and constantly monitor, reflect on, and adjust their motivation, cognition, and study behavior [37]. To better describe and measure a student's ability to regulate their learning process, [29] divided it into three subcomponents with two cognitive components (the use of cognitive strategies and the use of metacognitive strategies) and one non-cognitive component (resource management, including skills of time and study environment management, effort regulation, peer learning, and help seeking). A systematic literature review focused on online learning contexts found consistent evidence that resource management skills, especially time management skills and effort regulation skills, are predictive of performance [10]. While new technologies are creating novel measurement tools for these intangible qualities, the "ground truth" mostly comes from validated surveys.

2.2 Comparison of Different Data Sources

Previous work has examined combining various data sources for predictive analytics in higher education. For example, [2] combined institutional data, course performance data and LMS data to predict students' within-course success. However, there has been little work comparing the impact of various data sources on student success. [3] compared the impact of different types of institutional variables, including demographic variables, prior academic achievement, student majors, and academic achievement in college courses on predicting graduation and re-enrollment rates. [36] compared the impact of virtual learning environment (VLE) data, course assessment data, and a demographic variable on predicting whether a student's performance will drop in a course and whether a student will pass or fail a course. They generally found that using VLE data in conjunction with assessment data was seemingly better than using either alone. In what is perhaps the closest study to ours, [35] compared the impact of learning behavioral features, student background, and non-cognitive features measured

by a socio-emotional skill assessment on predicting within-course success. Our study differs from theirs in that we look at long-term outcomes as well as short-term outcomes, we analyze the fairness of predictive models, and we fit models that span across several courses.

2.3 Fairness of Predictive Analytics in Education

In recent years, the fairness and biases of machine learning algorithms and systems have developed into a focused research area in the general machine learning research community¹. Research efforts encompass developing statistical measures of fairness, evaluating existing algorithms/systems, and correcting for biases in algorithmic pipelines, among others. As fairness is a concept rooted in a variety of disciplines, it has been a consensus that there is no single “correct” definition of fairness. Rather, what is fair is highly dependent on the specific application scenarios [6]. As such, contextualizing the fairness research in different fields is critical to improving real-world applications.

In earlier education research, there has been a focus on heterogeneous effects across student subpopulations in the contexts of testing [34], observational studies [39] and program evaluation [31]. These earlier perspectives resonate with the current theme of fairness, but as the adoption of predictive analytics systems in education for high-stakes purposes has a comparatively shorter history, formalized research on fairness in such contexts has been somewhat limited. Among the handful of empirical papers that have directly evaluated this aspect of predictive analytics in education, [13] showed through a simulation study that misspecified student models in intelligent tutoring systems could leave “slow” learners at lower mastery levels than “faster” learners; [16] examined the ROC curves from MOOC dropout prediction models, and identified significant gaps between gender groups through slicing analysis; and [19] used college application materials to predict on-time graduation and, employing the same slicing analysis, concluded that their model could make fair predictions across five sociodemographic groups.

As [6] points out, while the biases of predictive systems may be attributed to unfair algorithms, they can also arise from biased data which “reflect historical prejudices against certain social groups, prevailing cultural stereotypes, and existing demographic inequalities”. Therefore, unlike the previous studies described above, this paper examines fairness as an attribute of *data sources* rather than of *algorithms*. We look at fairness with respect to between-groups differences in three metrics: accuracy, false positive rate, and false negative rate. These metrics are among the many fairness metrics that have been proposed in the literature [6]. For example, having an equal false negative rate between subgroups has been called “equality of opportunity” in the context of giving everyone an equal opportunity to receive a positive intervention (e.g., being part of the university’s honor roll for having a high GPA) [17].

3. DATA AND METHODS

3.1 Data Sources

¹<https://facctconference.org/>

Following Section 2.1, this study compares the three widely available data sources in higher education settings: institutional data, Canvas LMS log data, and survey data. Specifically, we drew the sample of all students who enrolled and received final grades in ten fully online, introductory STEM courses taught from 2016 to 2018 at a large, public research university in the United States. Six of the courses were in public health while the remaining four were distributed across biology, chemistry and physics. These courses were the subject of a large research project, where our research team administered a series of standard survey questions about students’ motivation, self-regulation and other psychological constructs before, during and/or after each course. Therefore, we had valid survey data across multiple courses. Also, looking at online courses ensured that LMS data can provide holistic representations of learning behavior. A total of 2,244 students were in the original dataset, and after data cleaning as described below in Section 3.2, the final sample size was 2,093. Traditionally underrepresented groups in STEM fields made up a large portion of the sample: 72% were female, 48% came from low-income families, 54% were first generation college students, 33% were underrepresented minorities (URM)², and 13% were transfer students.

3.2 Features and Outcomes

From each of the three data sources, we constructed a separate feature set in line with the literature. Table 1 gives a summary of these features. *Institutional features* included student demographics and academic achievement prior to college. *Click features* were derived from the LMS data and only included general measures of behavioral engagement to accommodate the variances in course design. Specifically, for each student in each course, we calculated the total number of clicks and total time spent over the first half of the course period. Time spent was calculated as the time lapse between adjacent click events. For the last click event of a student (with no subsequent event) or exceptionally lengthy lapses, we set a heuristic value of 90 seconds. The click counts and time spent were also broken down by categories, which were defined based on the URLs that click events pointed to, including “portal”, “tasks”, “content”, “communication”, “performance” and “miscellaneous.” Restricting to the first half of course period speaks to the scenario of early identification of at-risk students for instructors. *Survey features* included four constructs of self-regulated learning skills and self-efficacy [29] from pre-course surveys launched during the first week of these courses. The completion rates of these surveys ranged from 65% to 93% across the ten courses. All survey items were adapted from Motivated Strategies for Learning Questionnaire (MSLQ), a popular questionnaire to measure self-regulation skills in online learning [30]. Each of the four constructs was measured by the average of corresponding survey items (Table 2).

As for outcomes, we defined two success measures. *Short-term success* was defined as a binary indicator of whether a student’s final course grade was above the class median. Predicting this within-course outcome aligns with the needs of instructors to recognize struggling students in a timely manner [15]. Similarly, *long-term success* was defined as

²This includes African American, Hispanic, and Native American students.

Table 1: Features derived from the three data sources

Institutional	Click	Survey
Female	Total clicks	Effort regulation
Transfer	Total clicks by category	Time management
Low income	Total time	Environment management
First-gen	Total time by category	Self-efficacy
URM	(All above for the first 5 weeks)	
SAT total score		
High school GPA		

Table 2: Details of survey features. Each feature was calculated as the average of its associated items.

Feature	Items (5-point Likert scale)
Effort regulation	I often feel so lazy or bored when I study that I quit before I finish what I planned to do (reverse coded). I work hard to do well in courses even if I don't like what I am doing. When coursework is difficult, I give up or only study the easy parts (reverse coded). Even when course materials are dull and uninteresting, I manage to keep working until I finish.
Time management	I keep a record of what my assignments are and when they are due. I plan my work in advance so that I could turn in my assignments on time.
Environment management	I usually work in a place where I can read and work on assignments without distractions. I can ignore distractions around me when I study.
Self-efficacy	I'm certain I can master the skills taught in this course. I'm certain I can figure out how to learn even the most difficult course material. I can do almost all the work in class if I don't give up.

whether a student's average GPA in the year that followed the course was above the median of their classmates in that course. Predicting this longer-term outcome is of interest to academic advisors and institutional policymakers because it can help them make appropriate policy changes early in students' academic careers to increase student success and graduation rates [22]. We used class medians to construct these outcomes instead of certain grade thresholds in order to better compare short-term and long-term results.

We examined all possible combinations of the three feature sets ($2^3 - 1 = 7$) regarding their ability to predict the two success measures. Therefore, a total of 14 binary classification problems were formulated. To fairly compare the prediction performance of these feature sets, students with missing values on more than 25% of all the individual features in Table 1 were dropped, which accounted for the decrease in sample size from 2,244 to 2,093. All continuous numerical features were standardized by centering to the median and scaling according to the interquartile range (IQR) to better handle outliers. For the remaining missing values, we performed multivariate imputation, i.e., modeling each feature with missing values as a function of other features.

3.3 Predictive Models

For each classification problem, we employed three common classification algorithms: logistic regression, support vector machines (SVM), and random forests. Course-level leave-one-group-out cross validation was used. In other words, the algorithm looped through the ten courses, and in each iteration used one course as the test set for the model trained on the remaining nine courses. Predicted values for each course were then put together from the ten iterations to evaluate the overall prediction performance. As our focus was the predictive power of different feature sets instead of models, we chose the classifier that produced the highest F-score for each combination of feature set and outcome.

Because we used median splits to construct outcomes, class imbalance was not a concern and therefore no resampling was performed. The entire predictive modeling process was implemented using the scikit-learn Python library [28].

3.4 Evaluation

We evaluated the prediction results via three metrics. Accuracy measures the overall predictive power of the features used. False positive rate (FPR) reflects the probability of missing out "at-risk" students or "overplacing" students. False negative rate (FNR), on the other hand, captures the chances of "underplacing" students [32]. These metrics can shed light on potential consequences of using certain data source(s) in different applications. From there, we can compare the utility of different data sources in a holistic manner.

We further evaluated each data source's contribution to the fairness of prediction results. Fairness was conceptualized as the performance parity across student subpopulations when the prediction was performed on the entire student sample. Specifically, we focused on an array of historically disadvantaged subpopulations and compared each of them with a corresponding reference group on the three metrics. For example, we compared the accuracy, FPR and FNR within Latinx students with those within white students. Figure 1a and 1b plot the group size and outcome distribution of these selected groups, where the last group under each category was the reference group.

Statistically, we computed the following disparity metrics for each disadvantaged group g :

$$acc_disparity = acc_{ref}/acc_g \quad (1)$$

$$fpr_disparity = fpr_g/fpr_{ref} \quad (2)$$

$$fnr_disparity = fnr_g/fnr_{ref} \quad (3)$$

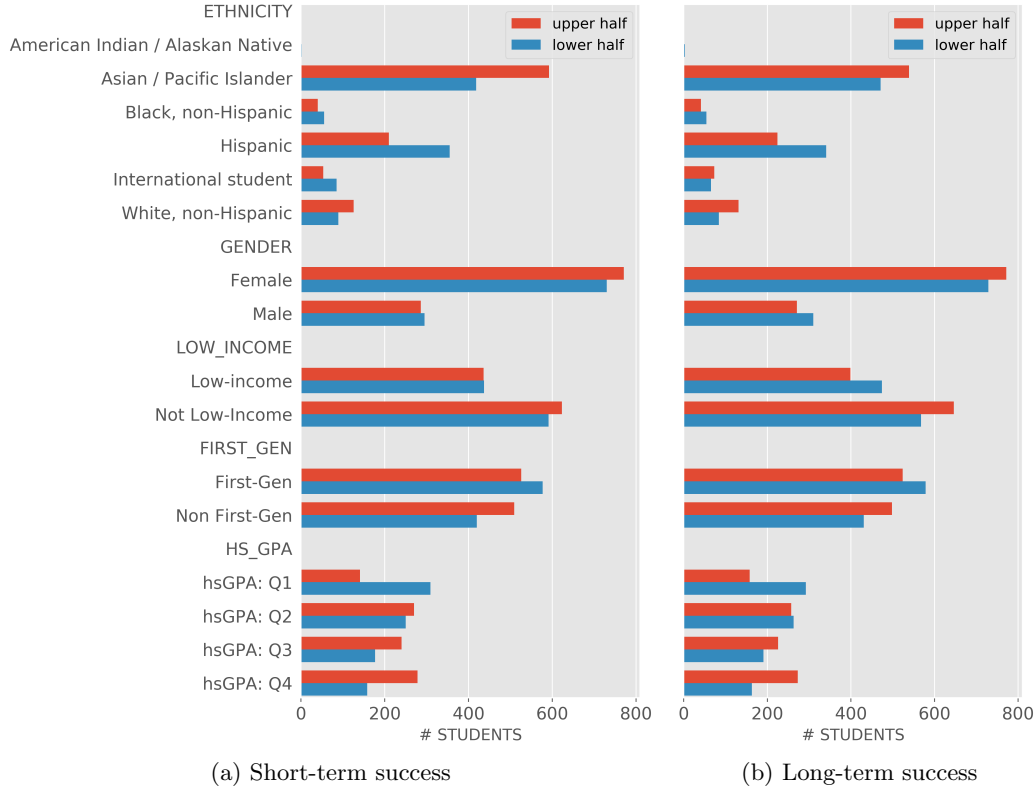


Figure 1: Outcome distribution within different student subpopulations. Short-term success: whether a student’s final course grade was above the class median. Long-term success: whether a student’s average GPA in the following academic year was above the class median.

and separately tested whether each of this disparities was significantly larger than 1 using one-sided two proportion z-test. The larger these ratios were, the more this student group was “discriminated against” by the prediction model. We used the less flexible one-sided test because of the consistent evidence that traditionally underrepresented groups experience more inequities than their counterparts in academic settings [1]. All these ratios combined would characterize the comparative utility of different data sources for fair predictions of college success.

4. PREDICTIVE UTILITY OF DIFFERENT DATA SOURCES

4.1 Overall Prediction Performance

Table 3 presents the prediction results on our full student sample across different feature and outcome combinations. In each column, the best-performing model is in bold to indicate which feature set(s) best predicted the corresponding outcome in the column header in terms of the given metric. Among the final sample of 2,093 students, 1,062 (50.7%) had short-term and 1,048 (50.1%) had long-term outcomes above their class median³. These numbers serve as the naïve baselines of prediction accuracy where all the students were simply predicted to be in the upper half (majority class).

³The slight deviation from 50% was due to the drop of students with too much missing information on predictors, as described in Section 3.2.

When the three data sources were used separately, institutional features and click features both achieved an overall accuracy of around 0.6 for either short-term or long-term outcomes, which was significantly higher than the baseline ($p < 0.001$ for all four cases). Specifically, institutional features appeared to be slightly more predictive of short-term success and click features predicted long-term success a little better, but neither of these comparisons was statistically significant. On the contrary, survey features had much weaker predictive utility because they predicted both outcomes with significantly lower accuracy than the worse of the other two features ($p < 0.001$ for short term and $p = 0.005$ for long term). When these feature sets were combined in different ways, we mostly saw improvement in the overall accuracy. The combination of institutional and LMS data led to the most noticeable accuracy increase in predicting both outcomes ($\Delta = 0.052, p < 0.001$ for short term and $\Delta = 0.037, p = 0.014$ for long term), evidencing complementary signals of student success in these two data sources. Survey data provided limited marginal utility as adding survey features to other feature sets never led to a statistically significant increase in accuracy and sometimes even had negative effects. However, the highest accuracy in predicting the short-term outcome was achieved when all three feature sets were used together.

Given the tradeoff between false positives and false negatives, overall best-performing feature sets did not necessarily

Table 3: Prediction performance on the entire student sample ($N = 2,093$). The best result in each column was in bold. Short: predicting whether a student’s final course grade was above the class median; long: predicting whether a student’s average GPA in the following academic year was above the class median.

Feature	Accuracy		FPR		FNR	
	Short	Long	Short	Long	Short	Long
Institutional	0.618	0.599	0.467	0.412	0.299	0.389
Click	0.602	0.613	0.485	0.385	0.313	0.389
Survey	0.534	0.557	0.599	0.385	0.336	0.502
Institutional+Click	0.670	0.650	0.351	0.330	0.310	0.370
Institutional+Survey	0.633	0.608	0.398	0.397	0.337	0.386
Click+Survey	0.609	0.604	0.431	0.457	0.353	0.335
Institutional+Click+Survey	0.675	0.638	0.348	0.402	0.303	0.323

have the lowest error rates. Among the three cases using a single data source, institutional features had both the lowest FPR and the lowest FNR for the short-term outcome ($p = 0.402$ for FPR and $p < 0.001$ for FNR compared to the second lowest). The same features also tied with click features for the lowest FNR in predicting the long-term outcome, while the latter led to the lowest FPR in the long term (tied with survey features). Combining these two data sources significantly lowered FPR ($\Delta = -0.116, p < 0.001$ for the short term and $\Delta = -0.055, p = 0.009$ for the long term) but not FNR. As for survey data, the patterns of error rates were more complicated than of overall accuracy. When used alone, survey features mostly led to higher error rates than the other two feature sets, except for FPR in the long term. On the other hand, adding survey features to other feature sets largely decreased FNR for long-term and FPR for short-term success predictions despite the fact that these metrics were exceptionally high in the case of using survey data alone.

4.2 Fairness of Predictions

Following Section 3.4, we computed and tested the extent to which each disadvantaged student subpopulation suffered discriminatory predictions (i.e., algorithmic bias) compared to their reference group under each combination of feature set and outcome. Figure 2a and 2b illustrate these results for short-term and long-term success prediction, respectively. Each cell colors a bias against a certain student subpopulation in a specific model. Darker cells suggest larger biases and crossed out cells represent those that were statistically significant ($p < 0.05$) after correcting for multiple testing within each background attribute. Subpopulations with fewer than 10 students were omitted as the error rates were less reliable.

Overall, there was no feature set that was entirely free from biased predictions. Across both outcomes, institutional features consistently led to higher FNR within various disadvantaged student subpopulations than within their peers. In other words, these students were more likely to be *underestimated* by the prediction model. This finding resonates with previous research that being aware of protected attributes (e.g., ethnicity) might induce identity-based biases in predictive analytics [6]. Adding other features to institutional ones alleviated some of these biases only in a marginal sense. That is, inclusion of institutional features seemed to largely determine the discriminatory behaviors of the model. Identity-blind LMS data was a fairer data source as the num-

ber of discriminated subpopulations was smaller. Compared to their reference groups, click features on their own significantly *overestimated* female students for both outcomes and Asian, Hispanic and first-generation college students for the long-term outcome. Survey data turned out to be neither accurate nor fair. When used alone, survey features led to significant biases against certain subpopulations across all metrics and outcomes. When combined with other feature sets, they did little to offset existing biases in most cases, except when they were used together with click features to predict long-term success. However, this latter case may suggest that survey data had equally low predictive utility for long-term success across different student subgroups.

The plots also allowed for insights into the extent to which different student subpopulations were exposed to algorithmic biases across different scenarios. Ethnic minorities, students from low-income families and first-generation college students were more prone to underestimation. Female students were more likely to be overestimated than male students especially in the long term. Moreover, international students and students with lower high school GPAs suffered both more underestimation and less accurate predictions compared to their peers. Note that unlike other variables in the plots, high school GPA is an acquired attribute. Hence, our evidence of algorithmic bias implied that a student can be stigmatized due not only to their demographic attributes but to their past (academic) experience as well.

4.2.1 A Closer Look into Institutional Data

Reflecting on the consistent biases against disadvantaged student subpopulations when using institutional data, we also tested if removing a specific institutional feature (e.g., gender) would eliminate the bias against the corresponding disadvantaged group (e.g., female). Surprisingly, all the results looked qualitatively similar regardless of which feature we removed. This suggested the intersectionality of minority identities, i.e., a student from one disadvantaged group tended to have another disadvantaged characteristic as well. As such, simply removing individual background variables would not necessarily make the predictions fairer.

5. DISCUSSIONS

5.1 Reflections on the Results

Our results shed light on the predictive validity of different sources of student data on college success. Our overall results agree well with those of [35], where features from an

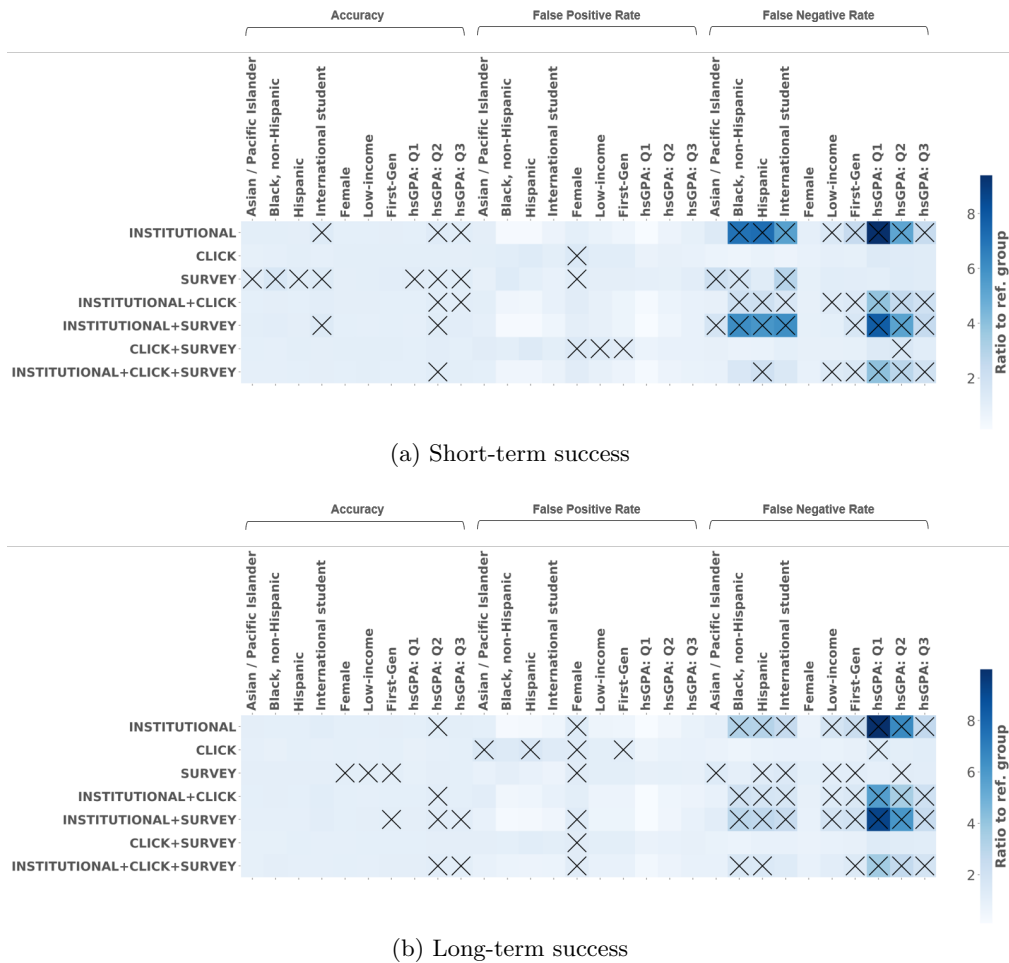


Figure 2: Illustration of prediction fairness. Each cell represents the algorithmic bias against a historically disadvantaged student subpopulation (compared to the corresponding reference group) in the specific scenario. Crosses represent statistically significant biases ($p < 0.05$) after correcting for multiple testing. Short-term success: whether a student’s final course grade was above the class median. Long-term success: whether a student’s average GPA in the following academic year was above the class median.

assessment of socio-emotional skills were least predictive of course success, which is similar to the ineffectiveness of our survey data. On the other hand, they found that models using institutional variables and clickstream features performed better and comparably to one another, as we did. They also discovered that combining clickstream behaviors with socio-emotional skills outperformed institutional data alone, which we also saw with the FNR for the long-term outcome. Interestingly, they did not find additional predictive utility of higher-level behaviors (sequential features) from clickstream data, which we did not further investigate.

The limited ability of pre-course survey data to accurately predict either short-term or long-term success may suggest that self-reported measures of self-regulated learning are not key factors of online learning processes or performance. However, as suggested by previous research [12], it may also suggest that students tend to overestimate their use of learning strategies in online courses. This is likely because students make estimations of their future behaviors based on memories of similar past events that are usually unreliable [21].

Thus, more research is needed to understand how to help students provide valid data of their learning skills as well as other psychological attributes in surveys [25].

When it comes to fairness, several interesting trends emerge. First, predictions using institutional data, which had the lowest FNR overall, were actually discriminatory when it comes to FNR for both outcomes. In particular, institutional data discriminated against students from underrepresented minority groups, low-income students, first-generation college students, and students with low high school GPA. This suggests that these models tend to disproportionately label students from these subpopulations as having below-median performance. In order to achieve higher overall accuracy, these models appear to be using a heuristic of classifying students as above or below median based on the majority class within the subpopulations that they belong to (see Figure 1). Therefore, one of the main sources of unfairness may just be the original class imbalance in different student subpopulations. When this imbalance results from historical inequities, the model will simply replicate those

inequities and produce unfair predictions.

On the other hand, we found that using click features tended to be fair with respect to FNR, but instead somewhat discriminatory with respect to FPR, for several student subpopulations. Contrary to the discrimination brought by institutional features, this form of discrimination could occur just because the model is blind to individual background. More specifically, students coming from different backgrounds may on average exhibit similar learning behaviors, but their likelihood to succeed might differ due to factors that correlate with their socio-economic status. Since the click features do not have access to students' background information, they may predict that students from disadvantaged backgrounds are likely to succeed at a disproportionately high rate.

One specific and possibly counterintuitive trend is seen when it comes to gender biases. While none of the feature sets discriminated against female students in terms of FNR, almost all of the feature sets discriminated against them in terms of FPR for at least one of the two outcomes. In fact, female students tend to have higher GPA than their male peers in the dataset (see Figure 1). This reinforces the inference that for institutional features, the models classify students into the majority class of their subpopulations in order to maximize accuracy. On the other hand, the fact that using only LMS and/or survey data is also biased against female students in terms of FPR might be due to something else. This suggests that female students might (a) exhibit different click behaviors and survey responses from men, which tend to be predictive of better performance; or (b) have different baseline levels of engagement (e.g., likelihood of clicking on LMS pages) independent of their likelihood of success. If the former is true, click behaviors and/or survey responses could act as a weak proxy for gender, even though gender is not encoded in these features.

5.2 Practical Implications

In general, prediction errors are inevitable, but it is important to be aware of and minimize potential misplacement that may result in severe negative consequences. Below, we discuss three major scenarios where prediction models are used for educational decision making and the implications of our findings in these cases.

First, higher education has a long history of screening applicants for desirable educational opportunities such as merit-based scholarships, where the award is based on the prediction of student future performance. In this case, underestimating student performance may limit their educational development. While institutional data is one of the most widely used data sources for these purposes, our results suggest that institutional data alone might be more likely to underestimate achievement of students from disadvantaged background as compared to their peers. Moreover, these systematic biases do not go away easily even when other common data sources are added. Therefore, it is important for policymakers to cautiously employ predictive analytics for selecting students since it may result in unfair exclusion of already disadvantaged students from critical educational opportunities and access to social mobility through education [18].

In community college settings, institutional data has also been used to evaluate students' readiness for college-level courses and assign students into remediation [32], as well as to understand the impact of remedial and preparatory courses on subsequent college success [24]. Put in this scenario, our results would suggest that students from historically disadvantaged subpopulations are more likely to be misplaced into remediation than their counterparts when they are actually capable of taking advanced courses. While remedial courses are designed to help academically underprepared students, they also increase students' cost and may delay student progression towards their degree goal [4]. For both this and the previous application scenarios, a potential algorithmic solution might be setting separate thresholds for different subpopulations to ensure fairness, as [20] suggested.

Finally, in the recent research and practice of online learning, LMS data have been commonly used to predict student performance and identify at-risk students [36]. Students who are identified as being at risk of low performance or dropout will often be placed into light-touch or optional academic support, such as receiving email reminders and tutoring services [11]. In this context, it might be more concerning to overestimate student performance and ignore students in need than to underestimate student performance and place them to educational resources that they could opt out of. Our findings indicate that compared to males, female students would be especially likely to experience overestimation and therefore would not receive academic resources that they need. In this case, incorporating institutional data into the prediction might not be as problematic in order to leave no student behind.

6. CONCLUSION

In this paper, we responded to the call for research to evaluate and compare the utility of common student data sources (i.e., institutional data, LMS data and survey data) for building predictive analytics applications in the context of higher education [14]. We aimed to find out what data sources and their combinations predicted short-term and long-term college success both accurately and fairly across different student subpopulations. Our results suggest that overall, institutional data and LMS data on their own have decent predictive utility for either instructors' or policymakers' needs to identify students in need. Using them together further strengthens that predictive power. Survey data alone poorly predicts student success and only marginally helps alleviate some of the prediction errors in the presence of other data sources. With regard to fairness, institutional data consistently leads to higher false negative rate (underestimation) within historically disadvantaged students subpopulations than within their peers. LMS data, on the other hand, tends to overestimate some of these disadvantaged groups (e.g., female students) more often than their counterparts and these biases would be overridden by institutional data when the latter is added. Survey data makes very limited contribution to fair predictions. Interestingly, all sources of student data tend to overestimate female students who perform better than male students on average in our case. Also, students with lower prior achievement are no less affected by underestimation than underrepresented demographic groups.

These results combined suggest that using multiple data

sources in college success prediction is beneficial for institutional stakeholders from both technical and ethical perspectives. Specifically, given the infancy and decent predictive utility of LMS data, institutions should feel encouraged to invest in the infrastructure to store, manage and analyze such data and integrate LMS-based behavioral measures into the routines of institutional research. On the other hand, utilizing multiple data sources still cannot guarantee fair predictions of college success especially for students who have less competitive academic records and who are historically disadvantaged in higher education. Therefore, it is advisable to combine the intelligence of experienced practitioners and data-driven applications for decision-making in the wild, in hopes of minimizing the risk that students are unfairly excluded from their optimal pathways due to biased algorithms or human judgement.

Our work has a few limitations which point to meaningful future work. First, the scope of our feature sets was limited and not representative of the full potential of different data sources. For example, for survey features we only used measures of self-regulation, but there are other psychological constructs that play equally important roles in learning processes. Therefore, our findings should be taken as a proof of concept in terms of systematically evaluating different data sources. Future work will extend the current piece to more comprehensive data sources that institutions have good control over [19, 3] and to broader feature sets informed by existing research. Second, while we briefly reflected on the prediction results and practical implications, we did not formally examine how the biases illustrated in Figure 2 permeate through the predictive analytics pipeline. Future work will examine this aspect more thoroughly, as well as how to convey these sources of bias to stakeholders for more prudent decision-making on student data usage.

7. ACKNOWLEDGMENT

This study is supported by the National Science Foundation (Grant Number 1535300).

8. REFERENCES

- [1] S.-A. A. Allen-Ramdiel and A. G. Campbell. Reimagining the pipeline: Advancing stem diversity, persistence, and success. *BioScience*, 64(7):612–618, 2014.
- [2] K. E. Arnold and M. D. Pistilli. Course signals at Purdue: Using learning analytics to increase student success. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, pages 267–270, 2012.
- [3] L. Aulck, D. Nambi, N. Velagapudi, J. Blumenstock, and J. West. Mining University Registrar Records to Predict First-Year Undergraduate Attrition. In *The 12th International Conference on Educational Data Mining (EDM)*, pages 9–18, Montréal, Canada, 2019.
- [4] T. Bailey, D. W. Jeong, and S.-W. Cho. Referral, enrollment, and completion in developmental education sequences in community colleges. *Economics of Education Review*, 29(2):255 – 270, 2010. Special Issue in Honor of Henry M. Levin.
- [5] R. Baker, D. Xu, J. Park, R. Yu, Q. Li, B. Cung, C. Fischer, F. Rodriguez, M. Warschauer, and P. Smyth. The benefits and caveats of using clickstream data to understand student self-regulatory behaviors: opening the black box of learning processes. *International Journal of Educational Technology in Higher Education*, 17:1–24, 2020.
- [6] S. Barocas, M. Hardt, and A. Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. <http://www.fairmlbook.org>.
- [7] G. Beattie, J.-W. P. Laliberté, C. Michaud-Leclerc, and P. Oreopoulos. What sets college thrivers and divers apart? A contrast in study habits, attitudes, and mental health. *Economics Letters*, 178:50–53, may 2019.
- [8] G. Beattie, J.-W. P. Laliberté, and P. Oreopoulos. Thrivers and divers: Using non-academic measures to predict college success and failure. *Economics of Education Review*, 62:170–182, feb 2018.
- [9] E. P. Bettinger, B. J. Evans, and D. G. Pope. Improving college performance and retention the easy way: Unpacking the ACT exam. *American Economic Journal: Economic Policy*, 5(2):26–52, may 2013.
- [10] J. Broadbent and W. L. Poon. Self-regulated learning strategies & academic achievement in online higher education learning environments: A systematic review. *Internet and Higher Education*, 27:1–13, 2015.
- [11] S. P. M. Choi, S. Lam, K. C. Li, and B. T. M. Wong. Learning analytics at low cost: At-risk student prediction with clicker data and systematic proactive interventions. *Journal of Educational Technology & Society*, 21(2):273–290, 2018.
- [12] M. K. DiBenedetto and H. Bembenuddy. Within the pipeline: Self-regulated learning, self-efficacy, and socialization among college students in science courses. *Learning and Individual Differences*, 23:218–224, 2013.
- [13] S. Doroudi and E. Brunskill. Fairer but not fair enough: On the equitability of knowledge tracing. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge - LAK19*, pages 335–339, Tempe, AZ, USA, mar 2019. ACM.
- [14] C. Fischer, Z. Pardos, R. S. Baker, J. J. Williams, P. Smyth, R. Yu, S. Slater, R. Baker, and M. Warschauer. Mining big data in education: Affordances and challenges. *Review of Research in Education*, 44:130–160, 2020.
- [15] D. Forteza, T. Harfield, J. Whitmer, and A. Dietrichson. What does it take to predict student risk? Evaluating LMS data to determine readiness for predictive modeling. Technical report, Blackboard Inc., 2017.
- [16] J. Gardner, C. Brooks, and R. Baker. Evaluating the Fairness of Predictive Student Models Through Slicing Analysis. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge - LAK19*, pages 225–234, Tempe, AZ, USA, 2019. ACM Press.
- [17] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.
- [18] R. Haveman and T. Smeeding. The role of higher education in social mobility. *The Future of Children*, 16(2):125–150, 2006.

- [19] S. Hutt, M. Gardner, A. L. Duckworth, and S. K. D'Mello. Evaluating Fairness and Generalizability in Models Predicting On-Time Graduation from College Applications. In *The 12th International Conference on Educational Data Mining (EDM)*, pages 79–88, Montréal, Canada, 2019.
- [20] J. Kleinberg, J. Ludwig, S. Mullainathan, and A. Rambachan. Algorithmic Fairness. *AEA Papers and Proceedings*, 108:22–27, 2018.
- [21] Q. Li, R. Baker, and M. Warschauer. Using clickstream data to measure, understand, and support self-regulated learning in online courses. *The Internet and Higher Education*, page 100727, 2020.
- [22] Y. Luo and Z. A. Pardos. Diagnosing University Student Subject Proficiency and Predicting Degree Completion in Vector Space. In *Proceedings of the Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, New Orleans, LA, USA, 2018.
- [23] W. Matcha, D. Gašević, N. A. Uzir, J. Jovanović, and A. Pardo. Analytics of Learning Strategies: Associations with Academic Performance and Feedback. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge - LAK19*, pages 461–470, Tempe, AZ, USA, 2019. ACM Press.
- [24] H. Nguyen, L. Wu, C. Fischer, G. Washington, and M. Warschauer. Increasing success in college: Examining the impact of a project-based introductory engineering course. *Journal of Engineering Education*, 2020.
- [25] J. L. Osterhage, E. L. Usher, T. A. Douin, and W. M. Bailey. Opportunities for self-evaluation increase student calibration in an introductory biology course. *CBE—Life Sciences Education*, 18(2):ar16, 2019.
- [26] Z. A. Pardos, Z. Fan, and W. Jiang. Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. *User Modeling and User-Adapted Interaction*, pages 1–39, feb 2019.
- [27] J. Park, R. Yu, F. Rodriguez, R. Baker, P. Smyth, and M. Warschauer. Understanding Student Procrastination via Mixture Models. In *Proceedings of the 11th International Conference on Educational Data Mining (EDM)*, Buffalo, NY, United States, 2018.
- [28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [29] P. R. Pintrich and E. V. De Groot. Motivational and Self-Regulated Learning Components of Classroom Academic Performance. *Journal of Educational Psychology*, 82(1):33–40, 1990.
- [30] P. R. Pintrich, D. A. F. Smith, T. Garcia, and W. J. McKeachie. A manual for the use of the motivated strategies for learning questionnaire (mslq). Technical report, Ann Arbor, MI, 1991.
- [31] M. C. Schippers, A. W. Scheepers, and J. B. Peterson. A scalable goal-setting intervention closes both the gender and ethnic minority achievement gap. *Palgrave Communications*, 1(1):1–12, 2015.
- [32] J. Scott-Clayton. Do High-Stakes Placement Exams Predict College Success? 2012.
- [33] D. Shapiro, A. Dundar, F. Huie, P. Wakhungu, A. Bhimdiwala, and S. Wilson. Completing College: A State-Level View of Student Completion Rates (Signature Report No. 16a). Technical report, National Student Clearinghouse Research Center, Herndon, VA, 2019.
- [34] R. L. Thorndike. Concepts of culture-fairness. *Journal of Educational Measurement*, 8(2):63–70, 1971.
- [35] J. Whitmer, S. S. Pedro, R. Liu, K. E. Walton, J. L. Moore, and A. A. Lotero. The Constructs Behind the Clicks. Technical report, ACT, Inc., 2019.
- [36] A. Wolff, Z. Zdrahal, A. Nikolov, and M. Pantucek. Improving retention: predicting at-risk students by analysing clicking behaviour in a virtual learning environment. In *Proceedings of the third international conference on learning analytics and knowledge*, pages 145–149, 2013.
- [37] C. A. Wolters. Self-regulated learning and college students' regulation of motivation. *Journal of educational psychology*, 90(2):224, 1998.
- [38] J. Xie, A. Essa, S. Mojarad, R. S. Baker, K. Shubeck, and X. Hu. Student Learning Strategies and Behaviors to Predict Success in an Online Adaptive Mathematics Tutoring System. In *Proceedings of the 10th International Conference on Educational Data Mining*, pages 460–465, Wuhan, China, 2017.
- [39] D. Xu and S. S. Jaggars. Performance Gaps Between Online and Face-to-Face Courses: Differences Across Types of Students and Academic Subject Areas. *The Journal of Higher Education*, 85(5):633–659, 2014.

The NAEP EDM Competition: On the Value of Theory-Driven Psychometrics and Machine Learning for Predictions Based on Log Data

Fabian Zehner
DIPF | Leibniz Institute for
Research and Information in
Education
fabian.zehner@dipf.de

Tobias Deribo
DIPF | Leibniz Institute for
Research and Information in
Education
deribo@dipf.de

Scott Harrison
DIPF | Leibniz Institute for
Research and Information in
Education
harrison@dipf.de

Daniel Bengs
DIPF | Leibniz Institute for
Research and Information in
Education
bengs@dipf.de

Carolin Hahnel
DIPF | Leibniz Institute for
Research and Information in
Education
hahnel@dipf.de

Beate Eichmann
DIPF | Leibniz Institute for
Research and Information in
Education
beate.eichmann@dipf.de

Nico Andersen
DIPF | Leibniz Institute for
Research and Information in
Education
andersen.nico@dipf.de

ABSTRACT

The *2nd Annual WPI-UMASS-UPENN EDM Data Mining Challenge* required contestants to predict efficient test-taking based on log data. In this paper, we describe our theory-driven and psychometric modeling approach. For feature engineering, we employed the Log-Normal Response Time Model for estimating latent person speed, and the Generalized Partial Credit Model for estimating latent person ability. Additionally, we adopted an n -gram feature approach for event sequences. For training a multi-label classifier, we distinguished inefficient test takers who were going too fast and those who were going too slow, instead of using the provided binary target label. Our best-performing ensemble classifier comprised three sets of low-dimensional classifiers, dominated by test-taker speed. While our classifier reached moderate performance, relative to competition leaderboard, our approach makes two important contributions. First, we show how explainable classifiers could provide meaningful predictions if results can be contextualized to test administrators who wish to intervene or take action. Second, our re-engineering of test scores enabled us to incorporate person ability into the estimation. However, ability was hardly predictive of efficient behavior, leading to the conclusion that the target label's validity needs to be questioned. The paper concludes with tools that are helpful for substantively meaningful log data mining.

Fabian Zehner, Scott Harrison, Beate Eichmann, Tobias Deribo, Daniel Bengs, Nico Andersen and Carolin Hahnel "The NAEP EDM Competition: Theory-Driven Psychometrics and Machine Learning for Predictions Based on Log Data" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 302 - 312

Keywords

Log Files, Psychometrics, Theory-Driven Feature Engineering, Process Data

1. INTRODUCTION

With the *2nd Annual WPI-UMASS-UPENN EDM Data Mining Challenge*,¹ the organizing consortium continued a young series of data competitions featured by the Educational Data Mining Society. The data challenge consisted in predicting students' behavior in a second test part using the log data produced in a first test part. The organizer's goal was to identify students who will act inefficiently by rushing through the second test half or not reaching the end of the test [19]. Another central, and noticeably constraining, secondary goal was that accurate classification should be reached as early as possible during test administration (i.e., with as little log data as possible) [19].

In this paper, we report details on our theory-driven psychometric contribution to the competition.² Opposed to data-driven analyses, a theory-driven one is characterized by identifying potential mechanisms at play and an according selection of methods, features, or both. The focus on a theory-driven feature-engineering access rather than some presumably more powerful deep-learning or other black-box methodology traces back to our team's psychometric background with strong experience in log data analysis. We believe that the theoretical understanding of underlying behavioral and cognitive processes that drive characteristics of test-taking behavior such as efficiency is crucial for build-

¹<http://tiny.cc/CompAIED> [2020-02-29]; also called *Nation's Report Card Data Mining Competition 2019*

²Our competition contributions have been submitted under the name *Team TBA* (Centre for Technology-Based Assessment | DIPF).

ing predictive models as requested in the given competition. Otherwise, the risk of integrating spurious associations into productive classifiers is high. Moreover, in the present paper, we provide evidence that the validity of the data challenge’s target label needs to be reassessed since we could show that students’ ability was hardly associated to the target label. Ability estimation was enabled by the re-engineering of scores from the log data—a unique contribution of the present paper. We suggest potential solutions for identified issues.

Efficiency can be defined as the characteristic of producing desired results without waste [13]. In the context of efficient test taking, this corresponds to successful test taking with minimum effort or time. Obviously, efficiency involves two components, namely goal-reaching and resource-saving. As we elaborate on in detail throughout the following sections, the competition’s operationalization of efficiency strongly emphasizes the latter component, but largely neglects the former. This consideration is emblematic and shows the value of a theory-driven and psychometric access to the matter. We regard log data that is captured during test administration as process data, which means it constitutes “empirical information about the cognitive (as well as meta-cognitive, motivational, and affective) states and related behavior that mediate the effect of the measured construct(s) on the task product” [7]. Thus, log data from assessment contexts is not just a by-product which is nice to have, but it carries relevant information and can be drawn on for purposes such as the one promoted in the competition.

With respect to classification performance, our competition contribution ended up in the top quarter of leaderboard submissions and was ranked eighth within the teams that submitted their code in time [20].

The paper first describes the setup provided by the competition organizers, then focuses on our approach for feature engineering as well as classifier training, and closes with reporting and discussing results on the classifier’s performance level as well as single features’ predictivity. The Conclusion Section elaborates on the definition of efficient test taking and discusses the state of the art for corresponding operationalizations. Please note that we use the terms *task* and *item* interchangeably here, in accordance with each community’s practice.

2. COMPETITION SETUP

2.1 Data

The competition data set [19] comes from the National Assessment of Educational Progress (NAEP), which is a US national assessment conducted across 4th-, 8th-, and 12th-grade students, including tests on a variety of subjects every two years. Specifically, the data set provided for analysis within the competition comes from the 2017 test for 8th-grade students in mathematics. The test comprised two test blocks (Block A and B) that were time limited to 30 min per block.

The NAEP 2017 mathematics assessment was digitally administered on tablet computers with keyboards [15]. The test items covered several domains such as algebra or geometry and were either presented as pure mathematics tasks or

as tasks applied in an everyday context. The items included stimulus material (text and/or figures) and either a list of responses to choose from (multiple choice), drag and drop response elements, or one or more text fields for constructed responses. The students were able to navigate between items within the same test block. For some items, a virtual calculator was available. The test environment also included a virtual drawing tool, which allowed students to freely draw and make handwritten annotations, a text-to-speech feature, which could be used to listen to the tasks’ written materials, and a help button offering the students information about the handling of the test environment. Before the first item, a screen with general instructions was presented, called *directions*. After the last item, a review screen was presented indicating the end of the test block. The students could navigate from the review screen back to the items and onward to the next test block. Students’ actions in this virtual environment (i.e., clicking on elements or keystrokes when typing a response) were recorded as log data events. The data provided 42 different events (e.g., **Enter Item**, **Click Choice**, **Back**). From this log data, it is possible to reconstruct the course of each student’s test completion behavior.

For the competition, the organizers arranged the log data into two sets. The first one was a training set, taken from Block A with log data for $n_{train} = 1232$ students across the whole 30 min of Block A. The data set was accompanied by the target label indicating whether students behaved efficiently in Block B (see next section). The second data set was an evaluation set for which efficiency labels had to be predicted by the contestants. This evaluation set comprised log data stratified into three conditions: i) $n_{10min} = 411$ students with 10 min of log data from the start of Block A; ii) $n_{20min} = 411$ students with 20 min of log data from the start of Block A; iii) $n_{30min} = 410$ students with complete 30 min of Block A. The competition organizers halved the evaluation set so that the leaderboard displayed the teams’ prediction accuracy on one half of the evaluation set, and the final evaluation was carried out on the remaining half. The training and evaluation sets consisted of 438,291 and 301,924 event logs, respectively.

2.2 Target Label: Efficiency

The competition organizers categorized the students into two groups. The value **True** indicated that a student completed Block B efficiently, while **False** indicated inefficient student test-taking behavior in Block B. Students were labeled efficient when they met two criteria: “1) being able to complete all problems in Block B, and 2) being able to allocate a reasonable amount of time to solve each problem” [19].

The definition of efficiency captures two key test-taking behaviors: students who go too slow, and as such fail to complete all the items in a block, and students who go too fast through the test, therefore not spending enough time on each question. Students who are inefficient through being too slow can easily be identified due to their failure to complete all tasks. However, for students going too fast, “a reasonable amount of time” can be difficult to operationalize. As such, the organizers chose to impose an arbitrary threshold for which students were evaluated on the total time taken on a task, with “the 5th percentile as the cut-off for the

‘reasonable amount of time’ [19]. This operationalization led to labeling 39.6% of the students in the training data as inefficient.

2.3 Evaluation Metrics

The objective of the competition was to develop a classifier model that would predict student efficiency. The prediction was evaluated against two key measures, the adjusted AUC and an adjusted kappa. The AUC stands for Area Under the Curve and comes from ROC analysis [4]. It compares the false positive rate to the true positive rate of the model, measuring how well the model predicts the correct outcome versus an incorrect prediction. A value of $AUC \leq .5$ would indicate a model performing no better than random chance. As such, the competition used an adjusted AUC measure, $AUC_{adj} = (AUC - 0.5) * 2$.

The second measure, kappa, also captures classifier performance by comparing how much two raters agree in classifying a given set of data beyond chance. Conceptualized by Cohen [3], it compares the observed accuracy to the expected accuracy between two classifiers. As such, the value of kappa needs to be above zero to indicate performance above random chance. The competition utilized an adjusted kappa value, κ_{adj} , in that they set the lower limit of kappa to 0. For the evaluation of the models within the competition, an aggregated score was made from AUC_{adj} and κ_{adj} .

3. METHODS

In this section, we first describe a data transformation step of splitting the three temporal conditions for feature extraction and training. This turned out to be essential for achieving appropriate classifier generalizability to the test set. Next, we describe our feature engineering as well as restrictive feature selection, and we close the section with outlining how the strings were pulled together for building an ensemble classifier for prediction.

All statistical analyses have been carried out using *R 3.6.1* [16], with the package *mlr 2.17.0* [2] for machine learning, *TAM 3.3-10* [18] for item difficulty and person ability estimation, and *LNIRT 0.4.0* [6] for item time intensity and person speed estimation.

3.1 Improving Generalizability by Separating Conditions

Our early submissions of predictions to the leaderboard revealed that the classifiers’ performance—though evaluated by stratified, repeated cross-fold validation—would always decrease substantially when being evaluated on the test set. That is, the generalizability of these classifiers to the test set was low, even when cross-validations testified to stable out-of-sample classification.

The primary reason that we identified was that the training set contained 30 min of log data, whereas the test set was split into three conditions with only the first 10 min, 20 min, or the full 30 min of log data available (see Section 2.1). Obviously, it is reasonable that feature realizations and their indication for one class vary over (testing) time. As an example, the time students take to work on single tasks does not only vary by task characteristics, but is also

influenced by the task’s position within the test. Another example is the log event of the timeout screen that limits students’ time to 30 min. Naturally, this event is reasonably predictive, but while it is available in the 30 min condition, it is not in the 10 min or 20 min condition. Therefore, training sets for each condition were necessary for the classifiers to generalize more properly to the test set.

For this purpose, we created three data sets: (i) the first 10 min of log data from the 10, 20, and 30 min conditions for predicting test set cases with 10 min of log data, (ii) the first 20 min of log data from the 20 and 30 min conditions for test-set cases with 20 min of log data, and (iii) the full 30 min of log data for test-set cases with 30 min of log data. For feature extraction, we combined the respective training and test (sub)sets. This way, we maximized the available information for norm-referenced features and parameter estimation procedures. Since we employed supervised learning methods, the test sets were excluded from classifier training.

The result of splitting the conditions was that we constructed three classifiers for each learning method and set of features. Each case in the test set, however, was classified by only one model, determined by the condition the test case belonged to.

3.2 Feature Engineering

In this section, we describe the selection of engineered features of which some ended up in at least one of the base classifiers that formed the final ensemble bag. We start with the two crucial psychometric models used for estimating students’ speed and ability. Then we describe our approach of extracting features from log data and deriving simple indicators that we assumed would indicate efficient or inefficient test behavior, using the software package *LogFSM*. Finally, we describe the concept and operationalization of rapid guessing as well as an adopted technique for representing log event sequences.

3.2.1 Latent Test-Taker Speed

Efficient test taking as operationalized in the competition (see Section 2.2) is mainly characterized by test takers’ time handling. If a student went relatively quickly through the test (in Block B), they were labeled as inefficient. If a student spent too much time on some tasks (in Block B), they would not be able to complete all tasks and thus be labeled as inefficient, too. Therefore, the most evident feature is test-taker speed.

Test-taker speed can be inferred from the time spent on tasks in a test. However, the time spent on a task is determined by the characteristics of the task and the test-taker. On the one hand, task characteristics, such as complexity, require and evoke a shorter or longer time on task due to the task’s inherent *time intensity*. On the other hand, some test takers will have the tendency or skill to move faster through a test than others; this characteristic is called *test-taker speed*. Both time intensity and test-taker speed are not directly observable and can only be estimated as latent variables.

A model that allows the separation of time on task into item and person parameters is the *Lognormal Response Time*

Model [22]:

$$f(t_{ip}; \tau_p, \alpha_i, \beta_i) = \frac{\alpha_i}{t_{ip} \sqrt{2\pi}} \exp \left\{ -\frac{1}{2} [\alpha_i (\ln t_{ip} - (\beta_i - \tau_p))]^2 \right\} \quad (1)$$

Response time distributions take values in the positive reals and typically have long tails. The log-transformation hence is a sensible way to approximate normality and is expected to lead to better fit than a normal model on the raw response times [22]. The lognormal model takes three parameters into account and is based on the log-transformed time t_{ip} that person p spent at item i . Item time intensity β_i captures item i 's tendency to evoke more or less time spent for completing it. Test-taker speed τ_p is a person's tendency and ability to spend more or less time on item completion. Because some items will show more homogeneous time distributions than others, the dispersion parameter α_i estimates an item's discriminatory power.

The parameters of interest are estimated in a Bayesian framework using a Markov Chain Monte Carlo method with a Gibbs sampler [22, 6]. We used expected a posteriori (EAP) estimators of test-taker speed τ_p as features for predictive modeling.

3.2.2 Latent Test-Taker Ability

The provided log data did not include task scores, so scores were re-engineered based on the log data and information from released items available through the NAEP questions tool [5]. To do this, unique item identifiers were mapped to example items provided in the NAEP questions tool, a public query tool used to showcase NAEP questions. The mapping was verified by text-to-speech contents in the log data. From this, the correct responses to items could be coded for 14 of the 19 items included in the competition data set. Using the 14 scored items, we estimated an intermediate ability score for test takers. By identifying the top 100 test takers across the 14 items, we then used their responses to the remaining 5 unreleased items to identify the most likely correct answer, thus inferring the correct scoring for the data. With this complete set of scores, we applied a Generalized Partial Credit Model [14] for estimating person ability. Theoretically, such ability estimates together with the speed estimates should be reasonably predictive of efficiency as efficiency is defined by a trade-off between performance and effort (see Section 1). The model is represented by the following equation [14]:

$$P_{jk|k-1,k}(\theta_p) = \frac{\exp[a_j(\theta_p - b_{jk})]}{1 + \exp[a_j(\theta_p - b_{jk})]} \quad (2)$$

The equation models the probability of a person p with the latent ability θ_p to respond to an item j by choosing the k th response category. In this model, subsequent response categories are ordered by their difficulty. The parameter b_{jk} represents the difficulty of an item's response category and a_j constitutes the item discrimination (i.e., the degree to which the item is capable of distinguishing between more or less able test takers). We used Marginal Maximum Likelihood for estimating model parameters. For person ability, Weighted Likelihood Estimators [24] were used. This way, test-taker ability θ_p can be directly used as a feature for

predictive modeling.

3.2.3 Simple Indicators of Students' Work Process

The analysis of process indicators is based on the assumption that latent characteristics of a test taker can be inferred from attributes of their work process [7]. However, the creation of indicators is often retrospective, depends on the specific assessment system employed, and is based on plausibility and expert opinion about which indicators might be of potential interest for a particular research question (e.g., time on task, number of page visits, or switching between environments). With the intent to provide a tool to facilitate the creation of process indicators from log data, the software package LogFSM [9] has been developed that can be used in R. Instead of providing a list of generic indicators, LogFSM requires the formulation of one or multiple theoretical models that a test developer or researcher has about the work process in a task. Afterwards, LogFSM reconstructs a given set of log data according to the predefined theoretical model(s). Attributes of the reconstructed work process then serve as process indicators.

The procedure of LogFSM utilizes the concept of finite state machines [10]. The work process is decomposed into a finite number of states which represent sections of the theoretically defined response process. For example, a researcher who wishes to distinguish process components in a math assignment might define the states *Task Reading*, *Task Processing*, *Responding*, and *Reviewing* that could alternatively be collapsed into states of lower granularity like *Stimulus Processing* and *Task Answering*. Practically, states are identified by events that represent test-taker interactions with the assessment platform (i.e., log events). The occurrence of such events can serve as the conditions that must be met in order to change from one state to another one, which is called transition. The interpretation of an event might differ from state to state, which may result in differences as to whether or not a transition is triggered. Depending on the previous state of a test taker, for example, a radio button click event might be interpreted as a first-time response (*Responding*) or an edited response (*Reviewing*). In summary, the interpretation of states and state sequences is constituted by the interplay of visible components of the assessment system (e.g., texts, images), the possibilities for interactions (e.g., buttons, text fields), the contexts in which events take place (e.g., accessing a calculator before or after a response was given), and—most importantly—the predefined assumptions about test-taking behavior and cognitive operations (e.g., reading instructions, reconsidering an answer) [10].

Finally, process indicators can be derived as attributes of the reconstructed states (or the reconstructed sequence of states) from log data that contextualize test-taking behavior according to the theoretically assumed test-taking process. The integration of the characteristics of a task, the available log events, and the theoretical expectations about the test-taking behavior assign a substantive meaning to an indicator [10]. For example, an indicator that reflects how long a student actually spends reviewing and checking a particular response again can be defined as the total time in a state *Reviewing* aggregated over multiple revisits of the task and cleaned for the time in other states such as *Responding*.

For the competition’s data analysis, we specified five FSMs to represent different attributes of students’ work process. The states of these FSMs represented students’ on-screen page (26 states); attempting, processing or reviewing of one of the 14 multiple-choice tasks (46 states) and tasks with other response formats (19 states); students’ use of the text-to-speech tool (4 states); and their use of the calculator and the drawing tool (5 states). Figure 1 shows the last mentioned model as an example. We distinguished between having the calculator active (state *CalcOn*), having the drawing tool active (state *textit*), and both tools being inactive (state *textit*). Transitions between states were triggered by the log events described in Section 2.1. For example, the state *CalcOn* was transferred to the state *ToolsOff* when the calculator was closed. That is, when the student pressed the calculator button (*CloseCalculator*), the drawing tool was activated (*ScratchworkModeOn*), or the item was left (*ExitItem*). Vice versa, when the drawing tool was activated, students’ could not open the calculator, allowing for the modeling of distinct states. Self-transitions were specified to deal with, for example, double-clicks.

Several simple indicators were then derived as aggregated attributes of the reconstructed states or sequence of states. For example, the number of occurrences of the state *CalcOn* across items reflects how often a student opened the calculator during the assessment. A summary of the derived simple indicators and their descriptions is provided in Table 1.

3.2.4 Rapid Guessing

Compromised effort and persistence have been shown to be identifiable by investigating rapid guessing behavior [25]. The concept of *rapid guessing behavior* is based on the assumption that the amount of time that a test taker spends on a task before responding is not sufficient to perceive the task and develop a serious solution [21]. A rapid guess is therefore defined as a response to a task with a response time below a certain threshold.

For the definition of the thresholds, multiple approaches are possible [26]. Following the competition’s operationalization of inefficient test-taking behavior [19], the present work identified task-specific response time thresholds for rapid guesses based on a 5th percentile cut-off value. This implies the assumption that the slowest 5 percent of test takers on each item showed rapid guessing behavior. This was in line with the competition’s definition of inefficient test-taking behavior and, thus, necessary for predicting the accordingly constructed target label. However, this is not state of the art and the Discussion Section reviews alternative approaches.

On the basis of the identified rapid guesses, a response matrix X_{pj} was constructed, indicating whether a response to task j by person p was observed and identified as a rapid guess. The entries in this matrix are specified as follows:

$$x_{pj} = \begin{cases} \text{NA} & \text{if no response is observed} \\ 0 & \text{if a response is observed \& a rapid guess} \\ 1 & \text{if a response is observed \& no rapid guess} \end{cases} \quad (3)$$

X_{pj} was then used to extract several rapid guessing indicators. The indicators encompass a dichotomous grouping-

Table 1: Simple Indicators Serving as Features or Used for Derived Feature Modeling

Indicator	Description
Time on Screen	Time a student spent on each task within the test. This included the directions, review, and help screens.
Tasks Attempted	A count of the number of tasks at which a student showed behavior indicating they were attempting to complete the task.
Tasks Completed	A count of the number of tasks a student had completed such that it could be scored.
Tasks Incomplete	A count of the number of tasks which a student attempted, yet left the response area with incomplete information; e.g., only placing 3 out of 4 drag-and-drop boxes into the response area.
Timeout	A binary variable indicating whether a student received the time-out screen, typically indicating that they failed to complete all tasks within the time limit of 30 min.
Reviews	A count variable indicating the number of times a student visited the review screen.
Too Fast	A count variable indicating the number of times a student was in the fastest 5% of test respondents for a given task.
Viewed/No Attempt	A count variable for the number of times a student viewed an item without interacting with the item in any meaningful manner.
Time on Directions	A time variable capturing the total amount of time spent on the directions screen.
Text to Speech	A count variable indicating the number of times a student utilized the text-to-speech feature.
Help	A count variable for the number of times a student opened the help dialogue to seek assistance.
Calculator	A count variable for the number of times a student opened the calculator feature.
Drawing Tool	A count variable for the number of times a student opened the drawing tool.

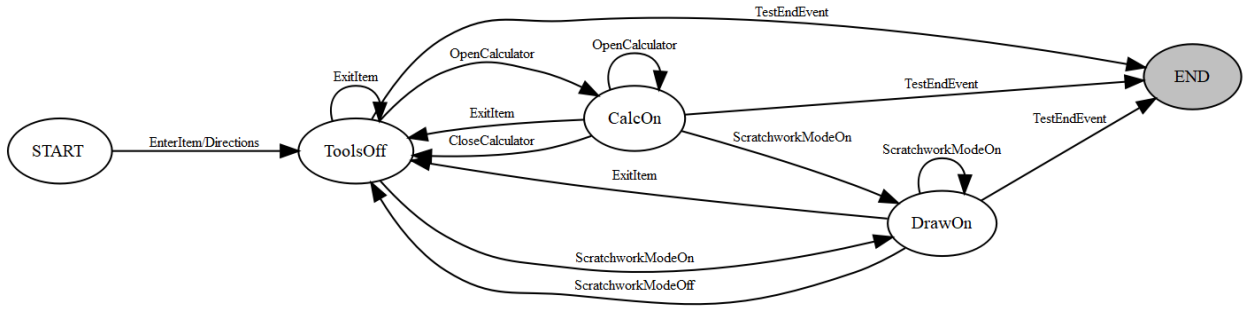


Figure 1: Exemplary Finite State Machine for Reconstructing Information from the NAEP Log Data

variable (whether a person showed at least one rapid guess), the sum of rapid guesses, and an estimation of a latent rapid guessing propensity [11]. For the estimation of the latent rapid guessing propensity, a Rasch model [17] was selected:

$$P(X_{pj} = 1) = \frac{\exp(\theta_p - \sigma_j)}{1 + \exp(\theta_p - \sigma_j)} \quad (4)$$

The Rasch model is similar to the GPCM presented in Section 3.2.2, just reduced to the dichotomous case and keeping the discrimination parameter constant. While the notation of symbols and indices is generally continued here, σ_j represents an item's difficulty (or propensity to evoke rapid guessing) and X_{pj} denotes the observed response correctness (or rapid guessing behavior), with $x \in \{0, 1\}$. For person parameter estimation, Expected A Posteriori estimates were used.

3.2.5 *n*-Grams of Log Events

The occurrence of certain log events can indicate behaviors or unobservable meta-cognitive, cognitive, or affective states of interest. This is also true for combinations of such. In the context of the competition, disengaged behavior might be a precursor or indicator for (later) inefficient test taking. For example, (a) whether a student uses the assessment system's drawing tool in a task that does not require its usage could be indicative of inefficient test taking as could be (b) the playing-around with the text-to-speech feature. For incorporating such predictive features, we adopted an approach by He and von Davier [8] that borrows techniques from natural language processing and information retrieval.

At the core of the procedure [8], a student's log events are considered as *n*-grams of a sequence. *n*-grams constitute all possible tuples of subsequent log events within a student's complete sequence of log events. For computational as well as sample size reasons, it is common to limit analyses to uni-, bi-, and trigrams. Hence, a sequence such as ACAD (representing four log events) would be decomposed into four unigrams ($2 \times \langle A \rangle$, $\langle C \rangle$, $\langle D \rangle$), three bigrams ($\langle AC \rangle$, $\langle CA \rangle$, $\langle AD \rangle$), and two trigrams ($\langle ACA \rangle$, $\langle CAD \rangle$). We decided to make each event task-specific; that is, the event **Draw** was captured together with the task ID, for example, **DrawTask4**. This way, events were contextualized. Varying by the 10, 20, and 30 min conditions, we obtained 7448, 13,482, and 17,553 *n*-grams, excluding sequences that occurred in less than 15 students' sequences.

Next, the frequency sf_{ij} of each *n*-gram *i* is computed for each student *j* (i.e., sequence frequency). These frequencies are then weighted by inverse sequence frequency (borrowing from the term *inverse document frequency*), $ISF_i = \log(N/sf_i)$, with *N* representing the total number of sequences, and log-normalized; that is $(1 + \log(sf_{ij})) * ISF_i$. This way, sequences occurring across many test administrations are scaled down in their importance and vice versa. Also, higher frequencies are dampened by the log-transformation.

The weighted *n*-gram frequencies can then be checked for their predictivity of, for example, efficiency, using a χ^2 -distributed statistic (details at [8, 12]). This revealed 841, 1259, and 1190 significantly predictive *n*-grams ($\alpha = .05$) for the respective condition.

In a last step, we compressed the selected features in a principal component analysis. Due to the need for a low-dimensional feature space (see Section 3.3), we extracted only a few components, retaining only 5% of the original information. This resulted in 6, 9, and 14 components, respectively, for the three conditions.

3.3 Feature Selection

We applied several different feature selection strategies. First, we used random forests to obtain features' importance for predicting students' efficiency in Block B. Second, we evaluated the accuracy of predictions using different combinations of features. Both strategies showed speed to be the most predictive feature in all conditions. However, the importance of the other features differed depending on the data set and combination of features.

Moreover, we frequently observed that if the addition of a feature improved the classification performance on the training data substantially (evaluated by stratified, repeated ten-fold cross-validation), it reduced the performance on the test data significantly. Thus, low-dimensional models were always to be favored over high-dimensional ones. For our final ensemble bag, the 10 and 20 min classifiers indeed turned out—with one exception—to work best with only one single feature: latent person speed. In the 30 min condition, more features were selected for the final prediction. For a list of the resulting features for all conditions, see the following Section 3.4.2.

Table 2: Three Sets of Base Classifiers

<i>Classifier Set (1): Speed & Test Completion</i>						
	ML	Speed	#Complete	#Incomplete	#TooFast	n-grams
10min	SVM	+				
20min	SVM	+	+			
30min	SVM	+	+	+	+	
<i>Classifier Set (2): Multiclass Speed & Test Completion</i>						
	ML	Speed	#Complete	#Incomplete	#TooFast	n-grams
10min	mSVM	+				
20min	mSVM	+				
30min	mSVM	+	+	+	+	
<i>Classifier Set (3): Speed, Test Completion, & n-Grams</i>						
	ML	Speed	#Complete	#Incomplete	#TooFast	n-grams
10min	JRip	+				+
20min	SVM	+				+
30min	SVM	+	+	+	+	+

3.4 Prediction

3.4.1 Harvesting More Information: Multi-Label Classification

The binary target label split students into efficient and inefficient test takers. However, the competition’s definition of *inefficient* behavior mixed two types of test takers: those who are going too fast and those who are going too slow. Since the two types have different feature realizations, the learning algorithms have to optimize towards at least two different conditions for the same class. Most algorithms’ optimization works better if they have less conditions to optimize for within each class.

Therefore, we used the latent test-taker speed feature for further splitting the inefficient category into *Going Too Slow* and *Going Too Fast*. This new target label with now three instead of two classes was used for one set of classifiers (see Section 3.4.2). For doing so, the latent speed estimated by the Lognormal Response Time Model (see Section 3.2.1) distinguished between students going too fast and going too slow. An analysis showed that substantial rapid guessing behavior started at a threshold of about $\tau = 0$ and, thus, optimally divided the two inefficient groups. The resulting target label identified about 23% of the test takers as going too fast and about 17% as going too slow, keeping the original share of 60% of efficient test takers.

3.4.2 Three Sets of Base Classifiers

For the final prediction, we created three sets of base classifiers that were to be merged in an ensemble bag. Each set followed a different idea, incorporated different features, and was trained by a different learning algorithm. In turn, each set contained three classifiers, with one of them tailored to the 10, 20, and 30 min condition, respectively. We experimented with different feature sets, learning algorithms (common ones such as support vector machines, AdaBoost, J48, neural nets, and others), and hyperparameters for each base classifier. Table 2 shows which features and learning algorithms were used in which classifier. Which features were included and which learning algorithm was employed was determined by resulting performance with respect to the leaderboard. Due to the unstable performance in the test set, no systematic hyperparameter tuning was carried out.

Our first set of classifiers used support-vector machines with a radial kernel and C-classification for all three conditions

(with $C = 1$, $\gamma = 1/n$, $\epsilon = 0.001$, shrinking). In the 10 min condition, only speed was used for the prediction. In the 20 min condition, the number of completed items was added. In the 30 min condition, all features that got through feature selection (except n-grams, on purpose) were incorporated: speed, number of completed items, number of incomplete items, and items completed too fast.

Our second set of classifiers was designed similarly to the first one, but with a multiclass support-vector machine and the multiclass label distinguishing going-too-slow and going-too-fast students (see Section 3.4.1). In the 10 and 20 min conditions, speed was the only predictor of importance according to the feature selection procedure. In the 30 min condition, again, all features (except n-grams) were incorporated.

Our third set of classifiers differed from the other two sets in that it incorporated one principal component of the n-grams of event sequences (see Section 3.2.5). Apart from that, the same set of features were used like in the second classifier set. The 10 min condition made use of a propositional rule learner instead of the otherwise employed support-vector machine. The rule learner’s parameters were set to $F = 3$ folds, $N = 2$ as the minimal weight, maximum error rate of included rules $\geq .5$, and pruning was used.

3.4.3 Ensemble Bag

The three described sets of classifiers were combined in a final ensemble classifier. We used the bagging approach by averaging probabilities of a condition’s three base classifiers, but favoring inefficient classifications. We chose to favor inefficient classification since our base classifiers produced not enough inefficient classifications. Therefore, we ended up with one ensemble bag of classifiers for the 10, 20, and 30 min condition each.

4. RESULTS

The final evaluation of our prediction resulted in $AUC_{adj} = 0.27$ and $\kappa_{adj} = .19$. In the leaderboard with all 82 competitors, this corresponded to rank 25, with several teams having submitted multiple results. In the final table, which only included 13 teams that submitted their code in time, our contribution was ranked eighth. The winner achieved $AUC_{adj} = 0.34$ and $\kappa_{adj} = .22$. The rather low performance values, even for the winners, were accompanied with corresponding differences between the test and evaluation set, resulting in substantial changes in the ranking and indicating rather unstable models being prone to changes in the evaluation data. This is in line with the wavering performance during testing we observed.

With respect to single features, two of them draw particular interest: test-taker speed and ability. Figure 2 shows their ROC curves. Obviously, the latent speed feature taken alone predicts efficient test taking noticeably well ($AUC_{adj} = 0.36$, $\kappa = .30$ in a single-feature support-vector machine³). In contrast, students’ ability does not capture a lot of relevant information for predicting efficient test taking ($AUC_{adj} = 0.16$, $\kappa = .07$ in a single-feature support-vector machine³).

³based on the 30 min training data and a stratified 10-times tenfold cross-validation

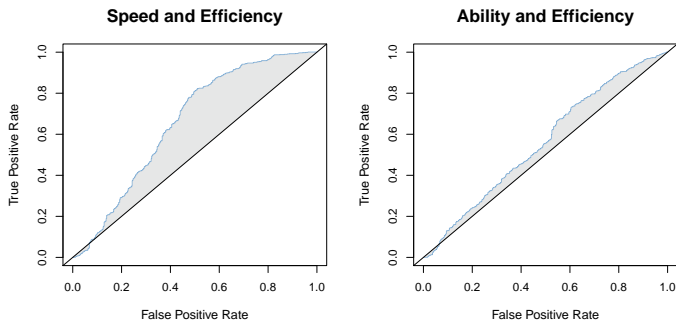


Figure 2: ROC Curves of Two Features: Speed (left) and Ability (right)

The large overlap of distributions between efficient and inefficient test takers for the ability feature further shows that the efficiency label does not contain much information about test takers’ ability (right part of Figure 3). There is a small difference in that inefficient students have lower ability values on average ($\Delta = -0.08$, Cohen’s $d = -0.20$). While the overlap of distributions appears somewhat similar for the speed feature (left part of Figure 3), the long right tail and prominence of faster inefficient test takers makes the feature space more easily separable. The effect size of the subgroups’ difference is remarkably higher ($\Delta = 0.12$, Cohen’s $d = 0.57$).

Finally, the feature space which is formed by ability and speed is plotted in Figure 4. The large majority of test takers builds an indistinguishable cloud. The other main message of this plot is, first, that very fast and less able test takers were consequently classified as inefficient in Block B. More surprisingly, second, a few test takers who were relatively fast, but answered correctly (and were thus estimated as relatively able) were classified as *inefficient* in Block B. It is possible that these students changed their behavior in the second test block. The other possibility is that the efficiency label classifies these instances erroneously as inefficient.

5. DISCUSSION

In this paper, we present a theory-driven psychometric modeling approach to predicting efficient test taking behavior in the context of the NAEP Data Mining Competition for 2019. The paper makes two important contributions, one to our understanding of the data, another to the structure of the competition.

The first major contribution is the value of theory-driven psychometric modeling for feature engineering. Referring back to Merriam Webster’s bipartite definition of efficiency as the characteristic of producing desired results without waste [13], it is interesting how task success is not incorporated into the competition’s conceptual specification of test takers. The data patterns mirror the lack of the desired results in the competition’s operationalization of the target label, demonstrating the prominence of speed as the sole determinant for the classification as efficient test taking. Remarkably, the outstanding speed feature serves as the only feature in some classifiers of our final ensemble bag that only

falls short of the winning contribution by $\Delta AUC_{adj} = .07$ and $\Delta \kappa = .03$. Empirically, ability did not provide any incremental increase in kappa or AUC beyond the speed feature. As a result, the ability feature was not included in any of the base classifiers after feature selection. It has to be noted that, at the theoretical level, the definition of efficiency only incorporates ability indirectly. That is the case because students who do not reach the end of the test cannot solve the corresponding items. Students who are going too fast are likely to fail as well. The resulting ability estimates, which are based on item success, hence, are indirectly incorporated in the efficiency label that is actually based on speeding criteria exclusively. Nevertheless, this indirect impact was not large enough for granting substantial predictivity to students’ ability for inferring their test-taking efficiency as specified by the competition.

It is apparent that the presented predictive modeling’s performance does not exceed a moderate level, if at all. This is similarly true for the competition winners. While behavioral predictions with temporal delay can always be expected to be weak, there seem to be multiple reasons inherent to the provided data set and challenge behind the moderate predictive classification performance. From our point of view, there are three major points that are worth following-up on in discussions. The most prominent one is the data reduction to twenty and ten minutes of log data for two thirds of the test data. The resulting leaderboard data evaluation was dominated by the secondary goal of predictions with less data. Also, since the different conditions shape the data and derived features quite differently, the training of classifiers had to be tailored to those.

The second important contribution is that the paper provides evidence that questions the target label’s validity. Using additional data sources from outside the information provided by the competition, we were able to re-engineer scores for estimating test taker ability. Importantly, feature selection led to excluding the ability feature, as it failed to be predictive of the efficiency label. This was a strong indicator for the suboptimal operationalization of efficiency.

This especially relates to the labeling of students as going too fast. To identify test takers spending a reasonable amount of time on a task, the competition organizers chose the 5th percentile of response times within an item as the threshold. Such a norm-oriented classification leads to labeling a fixed number of test takers as inefficient at each item, even when there are none or substantially less than 5 percent. Instead, criterion-based classification would be worthwhile. However, if corresponding criteria are not available, norm-oriented approaches would need to be combined with a dynamic threshold to be determined for each item, as the response time distributions of items typically differ considerably. The high ratio of 40 percent of students labeled as inefficient, which seems unreasonably high, is probably the result of this purely norm-based decision.

One option for identifying an appropriate threshold constitutes the visual inspection of distributions if little information about items are available. Often, response time distributions are bimodal. The first, very early peak is then typically associated with rapid guessing, while the second

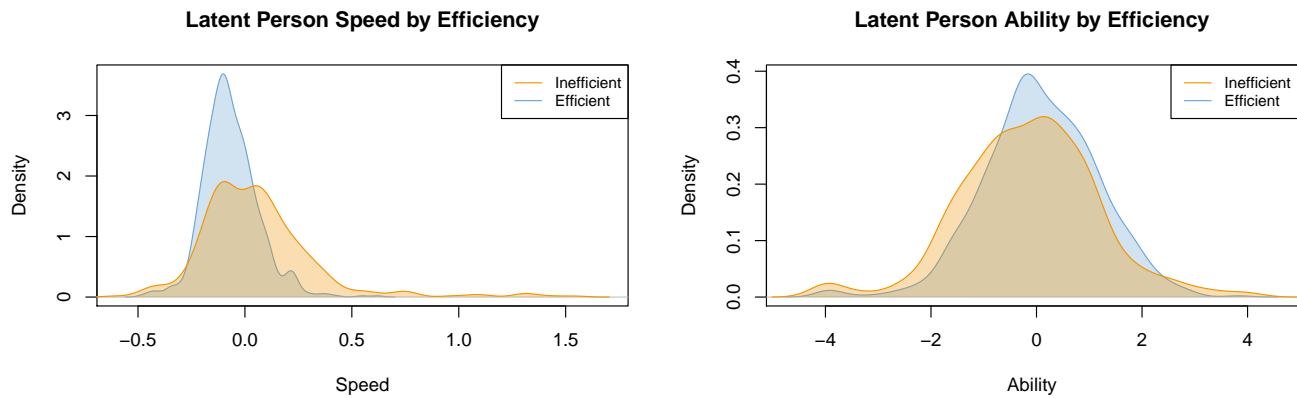


Figure 3: Distributions of Speed (left) and Ability (right), Separated by Efficiency

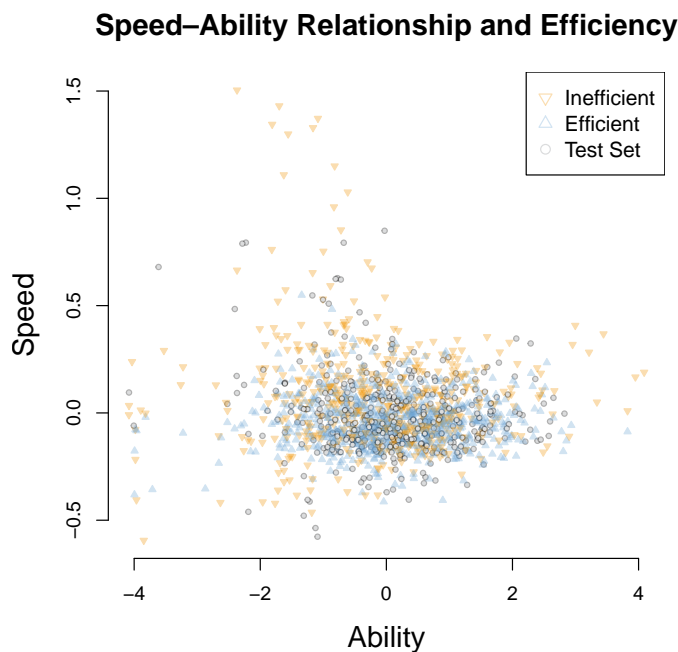


Figure 4: Scatter Plot of Speed and Ability, Distinguishing Efficient and Inefficient Test Takers

peak corresponds to the actual response time mean of those test takers who did not exhibit rapid guessing behavior. The threshold would be set after the obvious extinction of the first peak [27]. For setting an actually accurate threshold, methods that combine response time, item information, and response accuracy are considered state of the art. For an overview see for example [26]. Further, the identification of thresholds should be guided by contextual considerations, judging for example whether false-positives or false-negatives are more acceptable in the context of the test.

An additional area of interest was that the binary label for efficiency mixes two types of students within its inefficient value: going too fast and going too slow. This has implications for the learning algorithms that have to optimize their parameters towards two different conditions within one class. Moreover, from a substantive perspective, this mixes at least two types of students: those who are disengaged—thus, either rushing or meandering pointlessly through the test—and those who are too thoroughly working, poorly monitoring their progress, or who are just less able.

6. CONCLUSION

One of the central messages of the competition is that predictions of test-taking efficiency are highly dependent on the definition, measurement, and evaluation of efficiency itself. That is true for the presented approach, as well as for other competition entrants, as seen through the leaderboard test set evaluation phase. In such a case, and if classifiers are meant to be put into productive usage, it is even more important from our point of view to have comprehensible models. Imagine a hypothetical situation when a teacher sees a student being flagged on a dashboard after 20 min of testing. The flag indicates the risk for inefficient test taking later on, but we know that the flag’s accuracy is fairly low. It is vital that the teacher is informed about the basis of the flag’s decision criteria. As we have shown, the competition’s target label classified some of the most able students as inefficient who by ability are reasonably quick in completing the tasks. The consequences of a teacher going to a successful, engaged student and telling them they should aim at being more engaged or efficient in their test taking, would be reasonably disruptive. It can be assumed that such an

invasive and intrusive test administrator behavior would be counterproductive and decrease, rather than improve data quality. If however, the included features for predictions are transparent, known, and understandable, the teacher could communicate those and contextualize the flag accordingly. A risk of more powerful black-box deep-learning classifiers is that a small to medium share of more accurately classified cases does not necessarily outweigh the resulting obscurity of classification mechanisms. More generally, the effects of the invasive disruption of a test administrator proactively trying to motivate test takers on the standardization of the assessment setting need to be studied. Moreover, before using such a measure, classifiers would need to be checked for biases towards certain subgroups in order to still adhere to standards of standardized assessments [1]. Overall, we would recommend to refrain from using such predictions with low to moderate accuracy in productive assessments as long as the effects of changes in the test administration are unknown.

Instead, the discussion section gives some insights into what could improve the setup of a more proper training data set for predictions. Mainly, a more representative definition of efficiency might be necessary, one that reflects the current scientific state of the art which factors in students' ability. Furthermore, the described psychometric and theory-driven perspective, together with the referenced tools, can be helpful for mining log data from assessments at the large scale while retaining the individual perspective. With the illustrated software package LogFSM, for example, we were able to identify test takers who clearly showed consistent inefficient behavior, but were labeled as efficient, and vice versa. These observations are constrained by the fact that the log data of Block B was not available, yet served as the basis for the evaluation of the efficiency label. However, we think that the number of these cases is too large for being an effect of temporal instability only. We believe that these analyses combined with more innovative machine learning designs that the educational data mining community can provide are promising for further improving the predictions of test-taking efficiency.

7. LIMITATIONS

The paper already highlighted the presented study's limitations over the course of the different sections. On top of the challenges inherent to the data competition, this study's main limitation constitutes the employment of baseline machine learning. Moreover, speed and ability have been estimated separately, whereas a simultaneous estimation might have been possible as well [23]. The selection of feature sets and learning algorithms was optimized towards the test set which turned out to provide rather unstable evaluations. The conclusion of this paper is that the NAEP Data Mining Competition for 2019 provided an important opportunity to further develop complex conversations about how educational data mining and psychometric modeling can support data quality of assessments by identifying disengaged test taking behavior.

8. REFERENCES

- [1] AERA/APA/NCME. *Standards for educational and psychological testing*. American Educational Research Association, Washington, DC, 2014.
- [2] B. Bischl, M. Lang, L. Kotthoff, J. Schiffner, J. Richter, E. Studerus, G. Casalicchio, and Z. M. Jones. mlr: Machine learning in R. *Journal of Machine Learning Research*, 17(170):1–5, 2016.
- [3] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.
- [4] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.
- [5] N. C. for Educational Statistics. Naep questions tool.
- [6] J.-P. Fox, K. Klotzke, and R. K. Entink. *LNIRT: LogNormal Response Time Item Response Theory Models*, 2019. R package version 0.4.0.
- [7] F. Goldhammer and F. Zehner. What to make of and how to interpret process data. *Measurement: Interdisciplinary Research and Perspectives*, 15(3-4):128–132, 2017.
- [8] Q. He and M. von Davier. Analyzing process data from problem-solving items with n-grams: Insights from a computer-based large-scale assessment. In Y. Rosen, S. Ferrara, and M. Mosharraf, editors, *Handbook of Research on Technology Tools for Real-World Skill Development*, pages 750–777. IGI Global, Hershey, PA, 2016.
- [9] U. Kroehne. LogFSM: Analyzing log data from educational assessments using finite state machines. <http://logfsm.com/index.html>, 2019.
- [10] U. Kroehne and F. Goldhammer. How to conceptualize, represent, and analyze log data from technology-based assessments? A generic framework and an application to questionnaire items. *Behaviormetrika*, 45(2):527–563, Aug. 2018.
- [11] Y. Liu, Z. Li, H. Liu, and F. Luo. Modeling test-taking non-effort in mirt models. *Frontiers in Psychology*, 10:145, 2019.
- [12] C. D. Manning, C. D. Manning, and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.
- [13] Merriam-Webster. Efficiency. In *Merriam-Webster.com dictionary*. n.d.
- [14] E. Muraki. A generalized partial credit model: Application of an em algorithm. *Applied Psychological Measurement*, 16(2):159–176, 1992.
- [15] National Assessment Governing Board. *Mathematics Framework for the 2017 National Assessment of Educational Progress*. National Assessment Governing Board, Washington, DC, 2017.
- [16] R Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.
- [17] G. Rasch. *Probabilistic models for some intelligence and attainment tests*. University of Chicago Press, Chicago, IL, 1960/1980.
- [18] A. Robitzsch, T. Kiefer, and M. Wu. *TAM: Test analysis modules*, 2019. R package version 3.3-10.
- [19] Ryan Baker, Beverly Woolf, Irvin Katz, Carol Forsyth, and Jaclyn Ocumpaugh. Nation's report card data mining competition 2019. <https://sites.google.com/view/dataminingcompetition2019/home>, 2019.
- [20] Ryan Baker, Beverly Woolf, Irvin Katz, Carol Forsyth, and Jaclyn Ocumpaugh. Press release: 2019

- naep educational data mining competition results announced. <https://sites.google.com/view/dataminingcompetition2019/winners>, 2020.
- [21] D. L. Schnipke and D. J. Scrams. Modeling item response times with a two-state mixture model: A new method of measuring speededness. *Journal of Educational Measurement*, 34(3):213–232, 1997.
 - [22] W. J. van der Linden. A lognormal model for response times on test items. *Journal of Educational and Behavioral Statistics*, 31(2):181–204, 2006.
 - [23] W. J. van der Linden. A hierarchical framework for modeling speed and accuracy on test items. *Psychometrika*, 72(3):287, 2007.
 - [24] T. A. Warm. Weighted likelihood estimation of ability in item response theory. *Psychometrika*, 54(3):427–450, 1989.
 - [25] S. L. Wise. Rapid-guessing behavior: Its identification, interpretation, and implications. *Educational Measurement: Issues and Practice*, 36(4):52–61, 2017.
 - [26] S. L. Wise. An information-based approach to identifying rapid-guessing thresholds. *Applied Measurement in Education*, 32(4):325–336, 2019.
 - [27] S. L. Wise and X. Kong. Response time effort: A new measure of examinee motivation in computer-based tests. *Applied Measurement in Education*, 18(2):163–183, 2005.

Modeling Knowledge Acquisition from Multiple Learning Resource Types

Siqian Zhao*
Computer Science
University at Albany, SUNY
Albany, NY 12222 USA
szhao2@albany.edu

Chunpai Wang*
Computer Science
University at Albany, SUNY
Albany, NY USA
cwang25@albany.edu

Shaghayegh Sahebi
Computer Science
University at Albany, SUNY
Albany, NY 12222 USA
ssahebi@albany.edu

ABSTRACT

Students acquire knowledge as they interact with a variety of learning materials, such as video lectures, problems, and discussions. Modeling student knowledge at each point during their learning period and understanding the contribution of each learning material to student knowledge are essential for detecting students' knowledge gaps and recommending learning materials to them. Current student knowledge modeling techniques mostly rely on one type of learning material, mainly problems, to model student knowledge growth. These approaches ignore the fact that students also learn from other types of material. In this paper, we propose a student knowledge model that can capture knowledge growth as a result of learning from a diverse set of learning resource types while unveiling the association between the learning materials of different types. Our multi-view knowledge model (MVKM) incorporates a flexible knowledge increase objective on top of a multi-view tensor factorization to capture occasional forgetting while representing student knowledge and learning material concepts in a lower-dimensional latent space. We evaluate our model in different experiments to show that it can accurately predict students' future performance, differentiate between knowledge gain in different student groups and concepts, and unveil hidden similarities across learning materials of different types.

Keywords

knowledge tracing, domain modeling, tensor factorization, multi-view learning

1. INTRODUCTION

Both student knowledge modeling and domain knowledge modeling are important problems in the educational data mining community. In this context, student knowledge tracing and knowledge modeling approaches aim to evaluate students' state of knowledge or quantify students' knowledge in

the concepts that are presented in learning materials at each point of the learning period [15, 6, 51, 25, 53, 32, 14, 47]. Domain knowledge modeling, on the other hand, focuses on understanding and quantifying the topics, knowledge components, or concepts that are presented in the learning material [7, 12, 27]. It is useful in creating a coherent study plan for students, modeling students' knowledge, and analyzing students' knowledge gaps.

A successful student knowledge model should be personalized to capture individual differences in learning [51, 28], understand the association and relevance between learning from various concepts [42, 53], model knowledge gain as a gradual process resulting from student interactions with learning material [21, 38, 18], and allow for occasional forgetting of concepts in students [14, 32, 18]. Despite recent success in capturing these complexities in student knowledge modeling, a simple, but important aspect of student learning is still under-investigated: that students learn from different types of learning materials. Current research has focused on modeling one single type of learning resource at a time (typically, "problems"), ignoring the heterogeneity of learning resources from which students may learn. Modern online learning systems frequently offer students to learn and assess their knowledge using various learning resource types, such as readings, video lectures, assignments, quizzes, and discussions. Previous research has demonstrated considerable benefits of interacting with multiple types of materials on student learning. For example, worked examples can lead to faster and more effective learning compared to unsupported problem solving [33]; and enriching textbooks with additional forms of content, such as images and videos, increases the helpfulness of learning material [2, 1]. Ignoring diverse types of learning materials in student knowledge modeling limits our understanding of how students learn.

One of the obstacles in considering the combined effect of learning material types is the lack of explicit learning feedback from all of them. Some learning material types, such as problems and quizzes, are gradable. As students interact with such material types, the system can perceive student grade as an explicit feedback or indication of student knowledge: if a student receives a high grade in a problem, it is likely that the student has gained enough knowledge required to solve that problem. On the other hand, some of the learning materials are not gradable and their impact on student knowledge cannot be explicitly measured. For example, we cannot directly measure the consequent knowledge

*First two authors contributed equally to this work.

gain from watching a video lecture or studying an example.

As an alternative for quantifying student knowledge gain, the system can measure other quantities, such as the binary indication of student activity with a learning material or the time they spent on it. However, this kind of measure may result in contradictory conclusions [8, 23, 22]. For example, spending more time to study the examples provided by the system may both increase the student's knowledge, and at the same time, be an indicator of a weaker student, who does not have enough knowledge in the provided concepts. These weaker students may select to study more examples to compensate for their lower knowledge levels. Consequently, the knowledge gain of studying these auxiliary learning materials is usually overpowered by the student selection bias and is not represented correctly in the overall dataset.

A similar issue exists in the current domain knowledge models. The automatic domain knowledge models that are based on students' activities mainly model one type of learning material and ignore the relationship between various kinds of learning materials [17, 12]. Alternatively, an ideal domain knowledge model should be able to model and discover the similarities between learning materials of different types.

In this paper, we simultaneously address the problems of student knowledge modeling and domain knowledge modeling, while considering the heterogeneity of learning material types. We introduce a new student knowledge model that is the first to concurrently represent student interactions with both graded and non-graded learning material. Meanwhile, we discover the hidden concepts and similarities between different types of learning materials, as in a domain knowledge model. To do this, we pose this concurrent modeling as a multi-view tensor factorization problem, using one tensor for modeling student interactions with each learning material type. By experimenting on both synthetic and real-world datasets, we show that we can improve student performance prediction in graded learning materials, as measured by the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).

In summary, the contributions of this paper are:

- 1) proposing a personalized, multi-view student knowledge model (MVKM) that can capture learning from *multiple learning material types* and allow for occasional student forgetting, while modeling all types of learning materials;
- 2) conducting experiments on both synthetic and real-world datasets showing that our proposed model outperforms conventional methods in predicting student performance;
- 3) examining the resulting learning material and student knowledge latent features to show the captured similarity between learning material types and interpretability of student knowledge model.

2. RELATED WORK

Knowledge Modeling Student knowledge modeling aims to quantify student knowledge state in the concepts or skills that are covered by learning materials at each learning point.

Pioneer approaches of student knowledge modeling, despite being successful, were not personalized, relied on a predefined (sometimes expert-labeled) set of concepts in learning

material, did not allow for learned concepts to be forgotten by students, and modeled each concept independently from one another [19, 15, 36, 45]. Later, some student knowledge models aimed to solve these shortcomings by learning different parameters for each (type of) student [35, 51, 26], including decays to capture forgetting of concepts in learner models [39, 29, 31] and capturing the relationship between concepts that are present in a course [44, 21]. Yet, these models assume that a correct domain knowledge model, that maps learning material into course concepts, exists.

In recent years, new approaches aim to learn both domain knowledge model and student knowledge model at the same time [28, 20, 42, 48, 53, 18]. Our proposed model falls into this latest category as it does not require any manual labeling of learning materials, while having the ability to use such information if they are available. It is personalized by learning lower-dimensional student representations, allows forgetting of concepts during student learning by adding a rank-based constraint on student knowledge, and models the relationship between learning material.

Learning from Multiple Material Types In the educational data mining (EDM) literature, learning materials are provided in various types, such as problems, examples, videos, and readings. While there have been some studies in the literature on the value of having various types of learning materials for educating students [2, 8, 33], the relationship between these material types, and their combined effect on student knowledge and student performance is under-investigated.

Multiple learning material types have been studied in the literature in finding insights into different activity distributions or cluster patterns between high-grade and low-grade students [46, 49], have been used as contextual features in scaffolding or choosing among the existing student models [52, 43], have been added to improve existing domain knowledge models only for graded material types while ignoring student sequences [10, 13, 16, 30, 34, 41, 37], or have been classified into beneficial or non-beneficial for students [3]. However, to the best of our knowledge, none of these studies have explicitly modeled the contribution of various kinds of learning materials on student knowledge during the learning period, the interrelations among these learning materials, and their effect on student performance. The Bayesian Evaluation and Assessment framework found that assistance promoted students' long-term learning. More recently, Huang et al. discovered that adaptation of their framework (FAST) for student modeling by including various activity types may lead researchers to contradictory conclusions [23]. More specifically, in one of their formulations student example activity suggests a positive association with model parameters, such as probability of learning, while in another formulation this type of activity has a negative association with model parameters. Also, Hosseini et al. concluded that annotated examples show a negative relationship with students' learning, because of a selection effect: while annotated examples may help students to learn, weaker students may study more annotated examples [22]. The model proposed in this paper considers student interactions from multiple learning material types, mitigating over-estimation of student knowledge by transferring information from interactions with graded

material, while accounting for knowledge increase that happen as a result of student interaction with non-graded material.

3. MULTI-VIEW KNOWLEDGE MODELING

3.1 Problem Formulation and Assumptions

We consider an online learning system in which M students interact with and learn from multiple types ($r \in \mathcal{R}$) of learning materials. Each learning material type r includes a set of $P^{[r]}$ learning materials. A material type can be either graded or non-graded. Students' normalized grade in tests, success or failure in compiling a piece of code, or scores in solving problems are all examples of graded learning feedback. Whereas, watching videos, posting comments in discussion forums, or interacting with annotated examples are instances of non-graded learning feedback that the system can receive. We model the learning period as a series of student attempts on learning materials, or time points ($a \in \mathcal{A}$). To represent student interaction feedback with learning materials of each type r during the whole learning period \mathcal{A} , we use a $M \times P^{[r]} \times A$ three-dimensional tensor $\mathbf{X}^{[r]}$. The a^{th} slice of tensor $\mathbf{X}^{[r]}$, denoted by $X_a^{[r]}$, is a matrix representing student interactions with the learning material type r during one snapshot of the learning period. The s^{th} row of this interaction matrix $\mathbf{x}_{a,s}^{[r]}$ shows feedback from student s 's interactions with all learning materials of type r at attempt a ; and the tensor element $x_{a,s,p}^{[r]}$ is the feedback value of student s 's activity on learning material p of type r at learning point a .

We use the following assumptions in our model: (a) Each learning material covers some concepts that are presented in a course; the set of all course concepts are shared across learning materials; and the training data does not include the learning materials' contents nor their concepts. (b) Different learning materials have different difficulty or helpfulness levels for students. For example, one quiz can be more difficult than another one, and one video lecture can be more helpful than the other one. (c) The course may follow a trend in presenting the learning material: going from easier concepts to more difficult ones or alternating between easy and difficult concepts; despite that, students can freely interact with the learning materials and are not bound to a specific sequence. (d) As students interact with these materials, they learn the concepts that are presented in them; meaning that their knowledge in these concepts increases. (e) Since students may forget some course concepts, this knowledge increase is not strict. (f) Different students come with different learning abilities and initial knowledge values. (g) The gradual change of knowledge varies among different students. But, students can be grouped together according to how their knowledge changes in different concepts, e.g., some students are fast learners compared to others. (h) Eventually, a student's performance in a graded learning material, represented by a score, depends on the concepts covered in that material, student's knowledge in those concepts, the learning material difficulty/helpfulness, and the general student ability.

In addition to the above, we have an essential assumption (i) that connects the different parts of our model: a student's knowledge that is obtained from interacting with one learning material type is transferable to be used in other types of

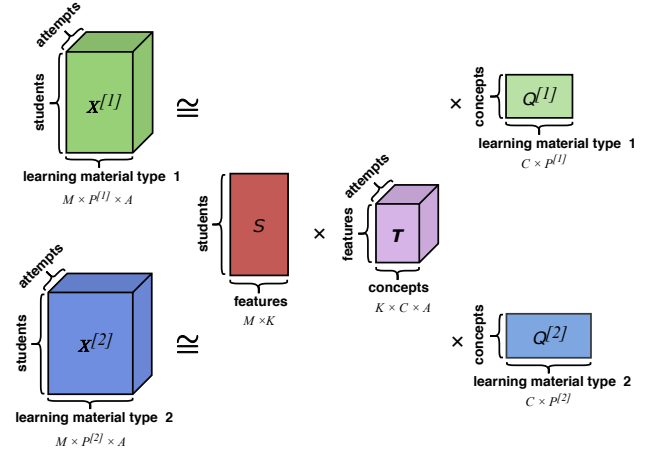


Figure 1: Decomposing student interaction tensors with two learning material types $\mathbf{X}^{[1]}$ and $\mathbf{X}^{[2]}$.

learning materials. In other words, students' knowledge can be modeled and quantified in the same latent space for all different learning material types. In the following, we first propose a single-view model for capturing the knowledge gained using one type of learning material (MVKM-Base) and then extend it to a multi-view model that can represent multiple types of learning materials.

3.2 MVKM Factorization Model

The Proposed Base Model (MVKM-Base). Following the mentioned assumptions in Section 3.1, particularly assumptions (a), (g), and (h), and assuming that students interact with only one learning material type, we model student interaction tensor \mathbf{X} as a factorization (n -mode tensor product) of three lower-dimensional representations: 1) an $M \times K$ student latent feature matrix S , 2) a $K \times C \times A$ temporal dynamic knowledge tensor T , and 3) a $C \times P$ matrix Q serving as a mapping between learning materials and course concepts. In other words, we have $\hat{x}_{s,a,p} \approx \mathbf{s}_s \cdot T_a \cdot \mathbf{q}_p$. Matrix S here represents students being mapped to latent learning features that can be used to group the students (assumption (g)). Tensor T quantifies the knowledge growth of students with each learning feature in each of the concepts while attempting the learning material. Accordingly, the resulting tensor from product $\mathbf{K} = \mathbf{S}\mathbf{T}$ represents each student's knowledge in each concept at each attempt.

To increase interpretability, we enforce the contribution of different concepts in each learning material to be non-negative and sum to one. Similarly, we enforce the same constraints on each student's membership in the student latent features. Since each student can have a different ability (assumption (f)) and each learning material can have its own difficulty level (assumption (b)), we add two bias terms to our model (b_s for each student s , and b_p for each learning material p) to account for such differences. To capture the general score trends in the course (assumption (c)), we add a parameter b_a for each attempt. Accordingly, we estimate student s 's score in a graded learning material p at attempt a ($\hat{x}_{a,s,p}$) as in Equation 1. Here, T_a is a matrix capturing the relationship between student features and concepts at attempt a , \mathbf{s}_s represents student s 's latent feature vector, \mathbf{q}_p shows material p 's concept vector.

$$\hat{x}_{s,a,p} \approx \mathbf{s}_s \cdot T_a \cdot \mathbf{q}_p + b_s + b_p + b_a \quad (1)$$

We use a sigmoid function $\sigma(\cdot)$ to estimate student interaction with a non-graded learning material, or graded ones with binary feedback:

$$\hat{x}_{s,a,p} \approx \sigma(\mathbf{s}_s \cdot T_a \cdot \mathbf{q}_p + b_s + b_p + b_a)$$

Modeling Knowledge Gain while Allowing Forgetting. So far, this simple model captures latent feature vectors of students and learning materials, and learns T as a representation of knowledge in students. However, it does not explicitly model students' gradual knowledge gain (assumption (d)). We note that students' knowledge increase is associated with the strength of concepts in the learning material that they interact with. As students interact with learning materials with some specific concepts, it is more likely for their predicted scores in the relevant learning materials to increase. With a Markovian assumption, we can say that if students have practiced some concepts, we expect their scores in attempt $a+1$ to be more than their scores in attempt a :

$$\mathbf{s}_s \cdot T_{a+1} \cdot \mathbf{q}_p - \mathbf{s}_s \cdot T_a \cdot \mathbf{q}_p \geq 0$$

However, this inequality constraint is too strict as the students may occasionally forget the learned concepts (assumption (e)). To allow for this occasional forgetting and soften this constraint, we model the knowledge increase as a rank-based constraint, that allows for knowledge loss, but penalizes it. We formulate this constraint as maximising the value for \mathcal{L}_2 in Equation 2. Essentially, this penalty term can be viewed as a prediction-consistent regularization. It helps to avoid significant changes in students' knowledge level since their performance is expected to transit gradually over time.

$$\mathcal{L}_2 = \sum_{j=1}^{a-1} \sum_{s,p} \log(\sigma(\mathbf{s}_s \cdot T_a \cdot \mathbf{q}_p - \mathbf{s}_s \cdot T_j \cdot \mathbf{q}_p)) \quad (2)$$

The Proposed Multi-View Model (MVKM). We rely on our main assumption (i) to extend our model to capture learning from different learning material types. So far, we have assumed that course concepts are shared among learning materials (assumption (a)). With the knowledge transfer assumption (i), all learning materials of different types will share the same latent space. Also, we represent student knowledge and student ability as shared parameters across all different learning material types. Consequently, for each set of learning materials of type $r \in \mathfrak{R}$, we can rewrite Equation 1 as:

$$\hat{x}_{s,a,p}^{[r]} \approx \mathbf{s}_s \cdot T_a \cdot \mathbf{q}_p^{[r]} + b_s + b_p^{[r]} + b_a$$

An illustration of this decomposition, when considering two learning material types, is presented in Figure 1. Note that we represent one shared matrix student S and one shared knowledge gain tensor T in both types of learning materials.

We can learn the parameters of our model by minimizing the sum of squared differences between the observed ($x_{s,a,p}^{[r]}$) and estimated ($\hat{x}_{s,a,p}^{[r]}$) values over all learning material types $r \in \mathfrak{R}$. For the learned parameters to be generalizable to unseen data, we regularize the unconstrained parameters using their L-2 norms. As a result, we minimize the objective function in Equation 3, in which $\gamma^{[r]}$ are hyper-parameters that represent the relative importance of different learning materials types.

λ_t and λ_s are hyper-parameters to control the weights of regularization term of T and S .

$$\begin{aligned} \mathcal{L}_1 &= \sum_{r,s,a,p} \gamma^{[r]} (\hat{x}_{s,a,p}^{[r]} - x_{s,a,p}^{[r]})^2 + \lambda_t \|T_a\|_F^2 + \lambda_s \|\mathbf{s}_s\|_F^2 \\ \text{s.t. } &\forall_{r,c,p} \quad q_{c,p}^{[r]} \geq 0, \quad \sum_c q_{c,p}^{[r]} = 1 \end{aligned} \quad (3)$$

Similarly, the knowledge gain and forgetting constraint presented in Equation 2 can be extended to the multi-view model. Eventually, we use a combination of the reconstruction objective function (Equation 3) and the learning and forgetting objective function (Equation 2) to model students' knowledge increase, while representing their personalized knowledge and finding learning material latent features, as in Equation 4. Note that, since our goal is to minimize \mathcal{L}_1 and maximize \mathcal{L}_2 , we use $-\mathcal{L}_2$ to minimize \mathcal{L} . To balance between the accuracy of student performance prediction and modeling student knowledge increase, we use a nonnegative trade-off parameter ω :

$$\mathcal{L} = \mathcal{L}_1 - \omega \mathcal{L}_2 \quad (4)$$

We use stochastic gradient descent algorithm to minimize \mathcal{L} in Equation 4. The parameters need to learn are students' latent feature matrix (S), dynamic knowledge in each concept at any attempt (T), importance of each concept in every learning material ($Q^{[r]}$), each student's general ability (b_s), each learning material's difficulty/helpfulness ($b_p^{[r]}$), and each attempt's bias (b_a).

4. EXPERIMENTS

We evaluate our model with three sets of experiments. First, to validate if the model captures the variability of observed data, we use it to predict unobserved student performances (Sec. 4.3). Second, to check if our model represents valid student knowledge growth, we study the knowledge increase patterns between different types of students and across different concepts (Sec. 4.4). Finally, to study if the model meaningfully recovers learning materials' latent concepts, we analyze their similarities according to the learned latent feature vectors (Sec. 4.5). Without loss of generalizability, although the model is designed to handle multiple learning material types, we experiment on two learning material types. Before the experiments, we will go over our datasets, and experiment setup.

Dataset	material type 1 (#)	material type 2 (#)	#stu	act. seq. len.	#rcds.	avg. sco.
Synthetic_NG	quiz (10)	discussion (15)	1000	20	19991	0.6230
Synthetic_NG2	quiz (10)	discussion (15)	1000	20	19991	0.6984
Synthetic_G	quiz (10)	assignment (15)	1000	20	19980	0.6255
MORF_QD	assignment (18)	discussion (525)	459	25	6800	0.8693
MORF_QL	assignment (10)	lecture (52)	1329	76	58956	0.7731
Canvas_H	quiz (10)	discussion (43)	1091	20	13633	0.8648

Table 1: Statistics for each datasets, where #stu is number of students, act. seq. len. is the maximum activity length, #rcds. is number of records that student interact with learning materials and avg. sco. is graded learning material's average score.

4.1 Datasets

We use three synthetic and three real-world datasets (from two MOOCs) to evaluate the proposed model. Our choice of real-world datasets is guided by two factors, aligned with

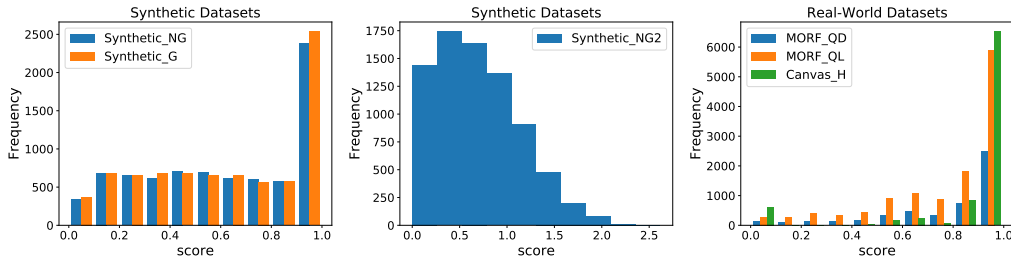


Figure 2: Histogram of graded materials' Scores in Synthetic Data and Real-World Data.

our assumptions: that they include multiple types of learning material, and that they allow the students to work freely with the learning material in the order they choose. In the real-world datasets, we select the students that work with both types of learning materials, removing the learning materials that none of these students have interacted with. General statistics of each dataset are presented in Table 1. Figure 2 shows score distributions of the graded learning material types in these datasets.

Synthetic Data. We generate three synthetic datasets according to two characteristics: (1) if both learning material types are graded vs. if one of them is non-graded (or has binary observations); (2) if the student scores are capped and their distribution is highly skewed vs. if the score distribution is not capped and less skewed.

For creating the datasets, we follow similar assumptions as to the ones made by our model. Expecting $P^{[1]}$ learning materials of type 1, and $P^{[2]}$ materials of type 2, we first generate a random sequence L_s for each student s , which represents the student's attempts on different learning materials. Considering C latent concepts, we then create two random matrices $Q^{[1]} \in \mathbb{R}^{C \times P^{[1]}}$ and $Q^{[2]} \in \mathbb{R}^{C \times P^{[2]}}$ as the mapping between the learning material and the C underlying concepts, such that the sum of values for each underlying learning material is one. For the student knowledge gain assumption, we represent each student's knowledge increase separately. Hence, we directly create a student knowledge tensor \mathbf{K} , instead of creating \mathbf{S} and \mathbf{T} , and multiplying them. To generate \mathbf{K} , we first generate a random matrix \mathbf{K}_1 that represents all students' initial knowledge in all C concepts. For generating the knowledge matrix in the next attempts (\mathbf{K}_a), we use the following random process. For each student s , we generate a random number α representing the probability of forgetting. If $\alpha > \theta$ (forgetting threshold), we assume no forgetting happens and increase the value in the knowledge matrix according to the learning material that the student has interacted with: $\mathbf{k}_{s,a} = \mathbf{k}_{s,a-1} + \beta \mathbf{q}_{L_s[a]}^{[r]}$. Here, β is a random effect of increasing and $L_s[a]$ is the learning material that student has selected to interact with at timestamp a . Otherwise ($\alpha < \theta$, or forget), we set $\mathbf{k}_{s,a,c} = \mathbf{k}_{s,a-1,c} - \text{rand}(0, \epsilon)$, for $\forall c \in C$. we use n-mode tensor product to build $\mathbf{X}^{[1]}$ and $\mathbf{X}^{[2]}$, where $\mathbf{X}^{[1]} = \mathbf{K}Q^{[1]}$, $\mathbf{X}^{[2]} = \mathbf{K}Q^{[2]}$. Finally, according to the student learning sequences L_s , we remove the "unobserved" values that are not in L_s from $\mathbf{X}^{[1]}$ and $\mathbf{X}^{[2]}$.

To create different data types according to the first characteristic above, for the graded learning material type r , we keep the values in $\mathbf{X}^{[r]}$. For the non-graded ones, we use the same process as above, except that in the final step we

set $x_{s,a,p}^{[r]} = 1$ according to the student sequence L_s . However, in many real-world scenarios, the score distribution of students is highly skewed especially towards higher scores (Figure 2 show it). To represent this skewness, in some of the generated datasets, we clip all $x_{s,a,p}^{[r]} > 1$ to 1.

Then, we create following three datasets according to above process: *Synthetic_G*, in which both learning material types are graded and scores are skewed; *Synthetic_NG*, in which one of the learning material types is graded and scores are skewed; and *Synthetic_NG2*, in which one of the learning material types is graded and scores are not skewed. We generate all synthetic data with 1000 students, $P^{[1]} = 10$ learning materials of type 1, $P^{[2]} = 15$ learning materials of type 2, $C = 3$ latent concepts, and maximum sequence length of 20 for students.

Canvas Network [11]. This is an online available dataset collected from various courses on the Canvas network platform¹. The available open online course data comes from various study fields, such as computer science, business and management, and humanities. For each course, its general field of study is presented in the data. The rest of the dataset is anonymized such that course names, discussion contents, student IDs, submission contents, or course contents are not available. Each course can have different learning material types, including assignments, discussions, and quizzes. We experiment on the data from one course in this system, with course id 770000832960975, which is in the humanities field (Canvas_H dataset). We use quizzes as the graded learning material type and discussions as the non-graded one.

MORF [4]. This is a dataset of the "educational data mining" course [5] at Coursera², available via the MOOC Replication Framework (MORF). The course includes various learning material types, including video lectures, assignments, and discussion forums. Students' history, in terms of their watched video lectures, submitted assignments, and participated discussions, in addition to the score they received in assignments, is available in data. In this course, we experiment with two datasets, each focusing on two sets of learning material types: one with assignments as the graded type and discussions as the non-graded type (MORF_QD), another with assignments as the graded type and video lecture views as the non-graded type (MORF_QL).

4.2 Experiment Setup

We use 5-fold student-stratified cross-validation to separate our datasets into test and train. At each fold, we use interaction records from 80% of students as training data. For the

¹<http://canvas.net>

²<https://www.coursera.org/>

rest (20%) of the students (target students), we split their attempt sequences on the graded learning material type into two parts: the first 50% and the last 50%. For performance prediction experiments, we predict the performance of the graded learning material type in the last 50%, given the first 50%. In order to see how the proposed model captures the knowledge growth, we do online testing, in which we predict the test data attempt by attempt (next attempt prediction). Eventually, we report the average performance on all five folds. For selecting the best hyper-parameters, we use a separate validation dataset. Our code and synthetic data are available at GitHub³.

4.3 Student Performance Prediction

In this set of experiments, we test our model on predicting student scores on their future unobserved graded learning material attempts. More specifically, we estimate student scores on their future attempts, and compare them with their actual scores in the test data.

4.3.1 Baselines

We compare our model with state-of-the-art student performance prediction baselines:

Individualized Bayesian Knowledge Tracing (IBKT) [24, 51]: This is a variant of the standard BKT model, which assumes binary observations and provides individualization on student priors, learning rate, guess, and slip parameters⁴.

Deep Knowledge Tracing (DKT) [38]: DKT is a pioneer algorithm that uses recurrent neural networks to model student learning, on binary (success/failure) student scores.

Feedback-Driven Tensor Factorization (FDTF) [40]: This tensor factorization model decomposes the student interaction tensor into a learning material latent matrix and a knowledge tensor. However, it only models one type of learning material, does not capture student latent features, and does not allow the students to forget the learned concepts. It assumes that students' knowledge strictly increases as they interact with learning materials.

Tensor Factorization Without Learning (TFWL): This is a model similar as FDTF, the only difference is TFWL does not have constraint that student knowledge is increasing.

Rank-Based Tensor Factorization (RBTF) [18]: This model has similar assumptions to FDTF. Except, it allows for occasional forgetting of concepts and has extra bias terms. Compared to MVKM, it does not differentiate between different student groups. It only uses student previous scores in graded learning materials to predict students' future scores, and it has a different tensor factorization strategy.

Bayesian Probabilistic Tensor Factorization (BPTF) [50]: This is a recommender systems model has a smoothing assumption over student scores in consecutive attempts.

AVG: This baseline uses the average of all students' scores for all predictions.

As mentioned before, one major issue in real-world datasets is their skewness, meaning that, on average, student grades are skewed towards a full (complete) score on quizzes/assignments. This skewness adds to the complexity of predicting an accurate score for unobserved quizzes: only using an overall average score will provide a relatively good estimate of

³<https://github.com/sz612866/MVKM-Multiview-Tensor>

⁴The code is from <https://github.com/CAHLR/pyBKT>

the real score. As a result, outperforming a simple average baseline is a challenging task.

The mentioned baselines all work on one type of learning material. Since our proposed MVKM model works with more than one learning material type, to be fair in evaluations, we run baseline algorithms in a multi-view setup. To do this, we aggregate the data from all learning material types and use that as an input to these baselines. In those cases, we add a "MV" to the end of their names. For example, FDTF_MV represents running FDTF on aggregation of student interactions with multiple learning material types. In addition, for knowledge tracing algorithms (BKT and DKT) which are designed for binary student responses (correct or incorrect), we modify their settings to make them predict numerical scores as described below. First, we binarize students' historical scores based on median score. Specifically, if the score is greater than the median, it will be set to 1, and 0 otherwise. Then, we use the probability of success generated by BKT and DKT as the probability of student receiving a score more than median score. Eventually, the numerical predicted scores can be obtained by viewing the probability output as the percentile of students' score on that specific question. Moreover, since these models require pre-defined knowledge components (KCs), we assume that each learning material is mapped to one KC in these models.

In addition to the above, we compare our multi-view model with its basic variation (MVKM-Base) using the data from graded materials only, and its multi-view variation without the learning and forgetting constraints (MVKM-W/O-P).

4.3.2 Performance Metrics and Comparison

In this task, our target is to accurately estimate the actual student scores. To evaluate how close our predicted values are to the actual ones, we use Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) between the predicted scores and the actual scores for students. Table 2 and 3 show the results of performance among different methods on synthetic data and real data, respectively. We can see that our proposed model outperforms other baselines on synthetic data, and has the best performance on real datasets in general.

MVKM-Base vs. Single Material Type Baselines.

Comparing MVKM-Base with other algorithms that use student scores only, shows us that MVKM-Base has consistently lower error compared to most baselines, in both synthetic and real-world datasets. This result demonstrates the ability of MVKM-Base in capturing data variance and validity of its assumptions for real-world graded data. Compared to AVG, MVKM-Base can represent more variability; compared to RBTF, the student latent features in MVKM-Base leads to improved results; compared to FDTF, the forgetting factor results in less error; and compared to BKT and DKT, modeling the learning material concepts in Q and having a rank-based constraint to enforce learning improves the performance. The only baseline algorithm that outperforms MVKM-Base in some setups is BPTF. Particularly, BPTF has a lower RMSE and MAE in Synthetic_NG and Synthetic_G datasets that are skewed. In real-world datasets, it performs better than MVKM-Base in MORF-QD dataset that is more sparse and has a slightly higher average score

Methods	Synthetic_NG		Synthetic_NG2		Synthetic_G	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
AVG	0.3084±0.0072	0.2820±0.0093	0.5059±0.0115	0.4005±0.0115	0.3070±0.0039	0.2811±0.0050
RBTF	0.2515±0.0126	0.2027±0.0081	0.3374±0.0234	0.2681±0.0146	0.2628±0.0113	0.2103±0.0080
FDTF	0.4906±0.0172	0.4410±0.0207	0.6588±0.0215	0.5529±0.0226	0.5041±0.0184	0.4537±0.0213
TFWL	0.5283±0.0168	0.4632±0.0178	0.6919±0.0132	0.5883±0.0156	0.5490±0.0053	0.5130±0.0076
BPTF	0.1675±0.0048	0.1256±0.0061	0.3454±0.0140	0.2589±0.0072	0.1825±0.0064	0.1381±0.0050
IBKT	0.4744±0.0118	0.4197±0.0140	0.6630±0.0122	0.5494±0.0152	0.4748±0.0076	0.4233±0.0098
DKT	0.2694±0.0275	0.1911±0.0241	0.4536±0.0404	0.3569±0.0413	0.2716±0.0209	0.2047±0.0178
RBTF-MV	0.2920±0.0069	0.2305±0.0078	0.4064±0.0213	0.3227±0.0147	0.2618±0.0155	0.2126±0.0130
FDTF-MV	0.4078±0.0168	0.3402±0.0167	0.5861±0.0211	0.4688±0.0135	0.4888±0.0112	0.4538±0.0131
TFWL-MV	0.4337±0.0139	0.3896±0.0133	0.6386±0.0161	0.5450±0.0194	0.5312±0.0137	0.4626±0.0145
BPTF-MV	0.1718±0.0037	0.1457±0.0055	0.3438±0.0158	0.2603±0.0120	0.1533±0.0055	0.1184±0.0044
IBKT-MV	0.4257±0.0142	0.3585±0.0155	0.6019±0.0124	0.4892±0.0165	0.4844±0.0068	0.4275±0.0089
DKT-MV	0.4278±0.0313	0.3613±0.0318	0.6399±0.0515	0.5320±0.0526	0.3390±0.0252	0.2892±0.0245
MVKM-Base	0.2007±0.1069	0.1498±0.0809	0.3026±0.0697	0.2273±0.0356	0.2097±0.0485	0.1565±0.0348
MVKM-W/O-P	0.1714±0.0089	0.1306±0.0089	0.2817±0.0316	0.2213±0.0245	0.1796±0.0345	0.1357±0.0190
Our Method (MVKM)	0.1388±0.0048	0.1049±0.0056	0.2221±0.0074	0.1739±0.0048	0.1532±0.0128	0.1171±0.0097

Table 2: Performance Prediction results on synthetic datasets, measured by RMSE and MAE, shown with variance in 5-fold cross-validation

Methods	MORF_QD		MORF_QL		CANVAS_H	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
AVG	0.2410±0.0227	0.1913±0.0161	0.2420±0.0108	0.1957±0.0067	0.0767±0.0121	0.0555±0.0040
RBTF	0.2711±0.0229	0.2132±0.0147	0.2572±0.0114	0.1980±0.0074	0.1571±0.0172	0.1235±0.0103
FDTF	0.3081±0.0437	0.2401±0.0329	0.3006±0.0194	0.2324±0.0151	0.1395±0.0259	0.0929±0.0119
TFWL	0.2750±0.0529	0.2003±0.0249	0.3090±0.3090	0.2237±0.0099	0.2377±0.0803	0.1186±0.0513
BPTF	0.2172±0.0128	0.1776±0.0082	0.2302±0.0068	0.1953±0.0048	0.1114±0.0120	0.0946±0.0082
IBKT	0.2756±0.0070	0.2281±0.0053	0.2646±0.0147	0.2174±0.0096	0.0856±0.0105	0.0692±0.0042
DKT	0.3169±0.0374	0.2498±0.0313	0.2859±0.0061	0.2158±0.0075	0.0911±0.0322	0.0616±0.0173
RBTF-MV	0.2814±0.0282	0.2177±0.0222	0.2624±0.0193	0.1977±0.0136	0.1484±0.0098	0.1171±0.0054
FDTF-MV	0.3138±0.0441	0.2453±0.0387	0.2398±0.0137	0.1866±0.0091	0.1149±0.0085	0.0907±0.0068
TFWL-MV	0.2919±0.0275	0.1975±0.0160	0.3222±0.0208	0.2178±0.0165	0.1748±0.0600	0.0784±0.0269
BPTF-MV	0.2615±0.0129	0.2286±0.0114	0.2313±0.0070	0.1960±0.0041	0.1452±0.0100	0.1343±0.0081
IBKT-MV	0.2774±0.0204	0.2177±0.0099	0.2904±0.0098	0.2137±0.0062	0.0834±0.0125	0.0425±0.0049
DKT-MV	0.2938±0.0310	0.2352±0.0236	0.2540±0.0065	0.2185±0.0047	0.079±0.0247	0.0496±0.0065
MVKM-Base	0.2242±0.0328	0.1669±0.0207	0.2277±0.0119	0.1724±0.0081	0.0666±0.0159	0.0411±0.0040
MVKM-W/O-P	0.2385±0.0196	0.1771±0.0104	0.2450±0.0145	0.1814±0.009	0.0649±0.0111	0.0388±0.0027
Our Method (MVKM)	0.2088 ± 0.0229	0.1603±0.0142	0.2150±0.0127	0.1654±0.0104	0.0613±0.0112	0.0362±0.0028

Table 3: Performance Prediction results on real-world datasets, measured by RMSE and MAE, shown with variance in 5-fold cross-validation.

compared to the other two. This shows that BPTF is better than MVKM-Base in handling skewed data. One potential reason is BPTF’s smoothing assumption, in contrast with MVKM-Base’s rank-based knowledge increase, that results in a more homogeneous score predictions for each student.

MVKM: Multiple vs. Single Material Types. Comparing MVKM’s results with MVKM-Base model, we can see that using data from multiple learning material types improves performance prediction results. It verifies our assumptions regarding knowledge transfer in different learning material types through the knowledge gain in shared concept latent space. This is given that in other models, e.g., all models except DKT in MORF-QD, adding different learning material types increases the prediction error. This error increase is particularly happening with BPTF model in real-world datasets and DKT model in synthetic ones. This shows that merely aggregating data from various resources, without appropriate modeling, can even harm the prediction results. This difference between MVKM and other baselines is in its specific setup, in which each learning material type is modeled separately, while keeping a shared knowledge space, student latent features, and knowledge gain.

Learning and Forgetting Effect. To further test the effect of our knowledge gain and forgetting constraint, we compare MVKM with MVKM-W/O-P, a variation of our

proposed model without the rank-based constraint in Equation 2. We can see that MVKM outperforms MVKM-W/O-P in all datasets. This shows that the soft knowledge increase and forgetting assumption is essential in correctly capturing the variability in students’ learning. Particularly, comparing MVKM-W/O-P’s results with MVKM-Base, the single-view version that includes the rank-based learning constraints, we can measure the effect of adding multiple learning material types vs. the effect of adding the learning and forgetting constraints in MVKM model. In CANVAS_H dataset, adding multiple learning material types is more effective than learning constraint, and in MORF datasets, realizing learning constraint is more important than modeling multiple types of learning materials. Nevertheless, they are not mutually exclusive and both are important in the model.

Hyper-parameter Tuning Using a separate validation set, we experiment with various values (grid search) for model hyper-parameters to select the most representative ones for our data. Specifically, we first vary the student latent feature dimension K in $[1, 5, 10, \dots, 40, 45]$, the question latent feature dimension C in $[1, 2, \dots, 9, 10]$, the penalty weight ω in $[0.01, 0.05, 0.1, 0.5, 1, 2, 3]$, the Markovian step m in $[1, 2, \dots, 10]$, and the learning resource importance parameter $\gamma^{[r]}$ in $[0.05, 0.1, 0.2, 0.5, 1, 2]$. Once we found a good set of hyper-parameters from coarse-grained grid search, we search the values close to the optimal values to find out the

best fine-grained values for these hyper-parameters. The best resulting hyper-parameter values for each dataset are listed in table 4. We use $\gamma^{[1]}$ as the trade-off parameter for graded learning material, $\gamma^{[2]}$ for another learning material. As we can see, in both the synthetic and real-world data, the learning and forgetting constraint is more important (larger ω) when having a non-graded learning material type. This shows that binary interaction data, unlike student grades (or scores), is not precise enough to represent the students' gradual knowledge gain in the absence of a learning and forgetting constraint. Also, comparing $\gamma^{[2]}$ in MORF_QD vs. MORF_QL we can see that the importance of video lectures is more than discussions in predicting students' performance.

Dataset	K	C	ω	$\gamma^{[1]}$	$\gamma^{[2]}$	η	m	λ_t	λ_s
Synthetic_NG	3	3	0.2	1	0.1	0.1	1	0.01	0.001
Synthetic_NG2	3	3	0.2	1	0.1	0.1	1	0.001	0.001
Synthetic_G	3	3	0.1	1	0.4	0.1	1	0.001	0.001
MORF_QD	39	5	1	1	0.05	0.1	1	0	0
MORF_QL	35	9	0.6	1	0.5	0.1	1	0	0
Canvas_H	28	7	2.0	1	0.5	0.01	1	0	0

Table 4: Hyperparameters of our model for each dataset

4.4 Student Knowledge Modeling

In this set of experiments, we answer two main research questions: 1) Can our model's learning and forgetting constraint capture meaningful knowledge trends across concepts for students as a whole? and 2) Are the individual students' knowledge growth representative of their learning? To answer these questions, we look at the estimated knowledge tensor of students ($\mathbf{K} = \mathbf{ST}$).

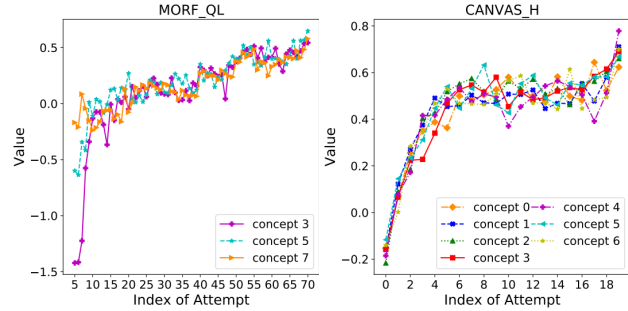


Figure 3: Average knowledge gain of concepts across all students.

To answer the first question, we check the average student knowledge growth on different concepts. Figure 3 shows the average knowledge of all students in different concepts (represented with different colors) during the whole course period (X-axis) for MORF_QL, and CANVAS_H datasets (MORF_QD has similar patterns as MORF_QL, we don't show it due to the page limitation). Notice that, for a clear visualization, we only show 3 out of 9 concepts from MORF_QL dataset in the figure. We can see that, on average, students' knowledge in different concepts increase. Particularly, in MORF_QL, the initial average knowledge on concept 3 is less than concepts 5 and 7. However, students learn this concept rapidly as shown by the increase of knowledge level around the tenth attempt. As the knowledge growth is less smooth in this concept, compared to the other two (e.g., the drop around the 15th attempt), students are more likely to forget it rapidly. Eventually, the average student knowledge in all concepts are close to each other. On

the other hand, in CANVAS_H, the average initial knowledge in different concepts are relatively close. However, students end up having different knowledge levels in different concepts at the end of the course, especially in concepts 0 and 4. Also, all six concepts show large fluctuations across the attempts. Overall, the students have a significant knowledge gain at the first few attempts and the knowledge gain slows down after that. This is aligned with our expectation on students' knowledge acquisition through out the course.

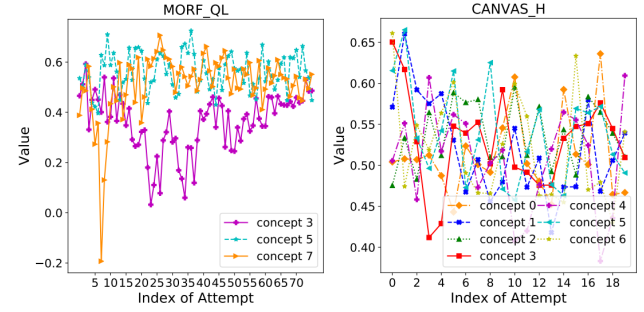


Figure 4: Average knowledge gain of each concept across all students.

To show the effect of the learning and forgetting constraint in MVKM, we look at the student knowledge acquisition in the MVKM-W/O-P model, that removes this constraint. The MVKM-W/O-P's average student knowledge in different concepts throughout all attempts is shown in Figure 4. We can see that despite its acceptable performance prediction error, MVKM-W/O-P's estimated knowledge trends are elusive and counter-intuitive. For example, many concepts (such as concept 3 in MORF_QL) show a U-shaped curve. This curve can be interpreted as the students having a high prior knowledge in these concepts, but forgetting them in the middle of the course, and then re-learning them at the end of the course. In some cases, such as concept 1 in CANVAS_H, students lose some knowledge and forget what they already knew, by the end of the course. This demonstrates the necessity of learning and forgetting penalty term in MVKM.

For second question, we check if there are meaningful differences between knowledge gain trends of different students. To do this, we apply spectral clustering on students' latent features matrix \mathbf{S} to discover different groups of students. Then, we compare students' learning curves from different clusters. The number of clusters is determined by the significance of difference on average performance in each cluster. We obtained 3 clusters of students for MORF_QD course, and 2 clusters for MORF_QL and CANVAS_H courses based on students' latent features from our model.

To see the differences in these groups, we sample one student from each cluster in each real-world dataset. Figure 5 shows these sample students' knowledge gain, averaged over all concepts, in datasets MORF_QD and MORF_QL (CANVAS_H is not showed due to the page limitation, it has similar patterns as MORF_QD). The figures show that different students start with different initial prior knowledge. For example, in MORF_QL, student #5 starts with a lower prior knowledge than student #100 and ends up with a lower final knowledge. Also, the figure shows that different knowledge gain trends across students, particularly in MORF_QD. For

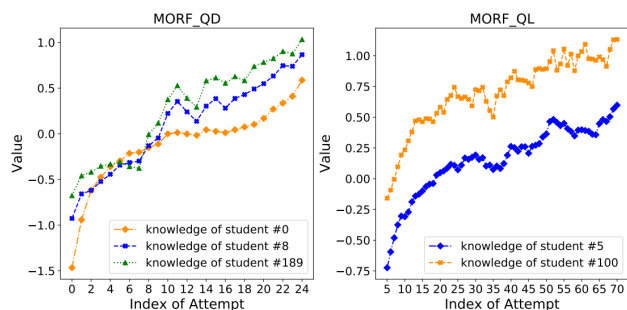


Figure 5: Sample students' knowledge gain across all concepts in two different courses.

example, student #0 starts with a lower prior knowledge than the other two students, but has a faster knowledge growth, and catches up with them around attempt 8. However, this student's knowledge growth slows down after a while and ends up to be lower than the other two at the end of course. To see if the quantified knowledge is meaningful, we compare student's knowledge growth with their scores. Students #0, #8, and #189 in MORF_QD have average grades 0.202, 0.636, and 0.909, in MORF_QL, #5 and #100 have average grades 0.9 and 0.98. This aligns with the knowledge levels shown in the figure. These observations show that MVKM can meaningfully differentiate between different students' knowledge growth.

4.5 Learning Resource Modeling

In this section, we evaluate our model on how well it can represent the variability and similarity of different learning materials. We mainly focus on two questions: 1) Are the learning materials' biases consistent with their difficulty levels? 2) Are the discovered latent concepts for learning materials (matrix $Q^{[r]}$) representative of actual conceptual groupings of learning materials in the real datasets?

Bias Evaluation. For the first question, since we do not have access to the learning materials' difficulty levels, we use average student scores on them, as a proxy for difficulty. As a result, we only use graded learning materials for this analysis. We calculated the spearman correlation between question bias captured by our model and average score of each question. The spearman correlation on MORF_QD is 0.779, on MORF_QL is 0.597, and on CANVAS_H is 0.960. We find that question bias derived from MCKM is highly correlated with average question score, where the lower the actual average grades are, the lower the bias values are learned.

Within-Type Concept Evaluation. For the second question, we would like to know how much the learning materials' discovered latent concepts resemble the real-world similarities in them. To evaluate the real-world similarities, we rely on two scenarios: 1) the learning material that are arranged closely to each other in the course structure, either in same module or in consequent modules, are similar to each other (course structure similarity); 2) the learning materials that are similar to each other have similar concepts and contents (content similarity). Since only one of our real-world datasets, MORF_QL, includes the required information for these scenarios, we use this dataset in the continuation of this paper. For first scenario, the course includes an ordered list modules, each of which include an ordered list of videos,

in addition to the assignments associated with each module.

For the second scenario, because our learning materials are not labeled with their concepts in our datasets, we use their textual contents (not used in MVKM) as a representation of their concepts. Particularly, we have subscripts for 40 video lectures, and text of questions for 8 quizzes. We note that if two learning materials present the same concepts, their textual contents should also be similar to each other. As a result, we build content-based clusters of learning materials, each of which containing the learning materials that are conceptually similar to each other. Specifically, to cluster the learning material according to their contents, we use Spectral Clustering on the latent topics that are discovered using Latent Dirichlet Analysis (LDA)[9] on the learning material's textual contents. In the same way, we can cluster the learning materials according to their discovered latent concepts by MVKM. Similar to the textual analysis, we use spectral clustering on the discovered $Q^{[r]}$ matrices to form clusters of learning materials. To do this, we first consider only one learning material type (the video lectures) and then move on to the similarities between two types of learning materials (both video lectures and assignments).

The results are shown in Figure 6 for within-type learning material similarity in video-lectures. Figure 6(a) shows the 8 clusters that were discovered using MVKM, and Figure 6(b) shows the 8 clusters that were discovered using video-lecture transcripts. Each cluster is shown within a box with a number associated with it. Each video-lecture is shown by its module (or week in the course), its order in the module sequence, and its name. For ease of comparison, we colored the video names according to their LDA content clusters. Looking at the LDA content clusters, we can see that although some lectures in same module fit in same cluster (e.g., videos 1, 2, 3, and 4 from week 7 are all in cluster 7), some of the lectures do not cluster with other videos in their module. For example, video 5 in week 7 is in cluster 2, with pioneer knowledge tracing methods. This shows that in addition to structural similarities, content similarities also exist in learning materials. Looking at MVKM clusters, we can see that the clusters mostly represent the course structure similarity: learning materials from same module are grouped. For example, all videos of week 3 are grouped in cluster 2. However, we can see that in many cases, whenever the structure similarity in clusters are disrupted, it is because of the content similarity in video lectures. For example, video 5 in week 7 that was clustered with pioneer knowledge tracing method in LDA content clusters is also clustered with them in MVKM clusters.

Between-Type Concept Evaluation. To evaluate MVKM's discovered similarities between different types of learning materials, we evaluate assignments' and video lectures' in MORF_QL. To do this, we build LDA-based clusters using assignment texts and video lecture transcripts. These clusters are shown in Figure 7(b). We also cluster the learning materials using spectral clustering on the concatenation of their $Q^{[r]}$ matrices (Figure 7(a)). Because the assignments bring more information to the clustering algorithms, the clustering results are different from the clusters of video lectures only. Similar to within-type concept evaluation results, we can still see the effect of both content and structure

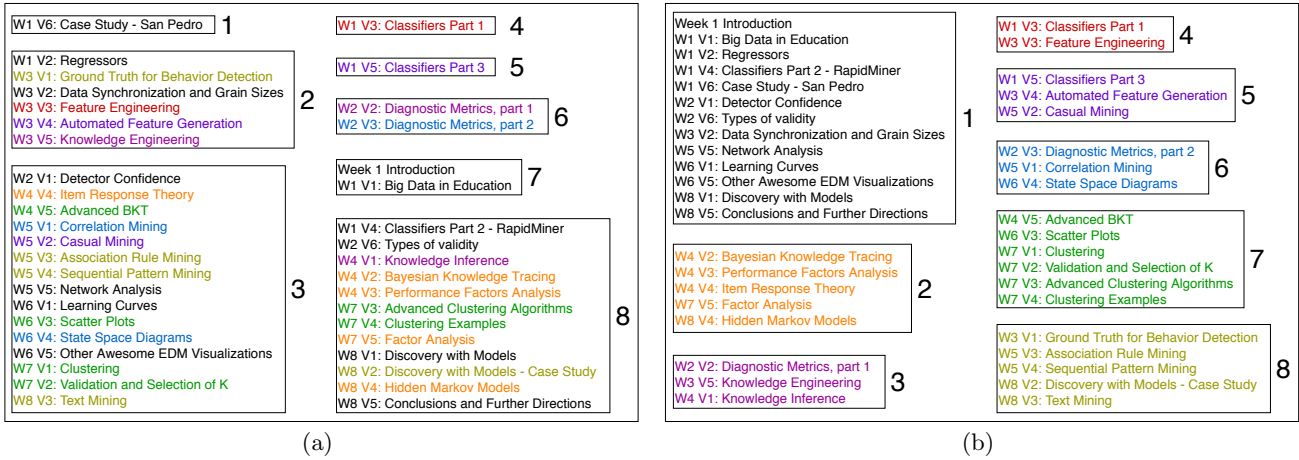


Figure 6: Clusters that were discovered by using MVKM (a), clusters discovered by using video-lecture transcripts (b).

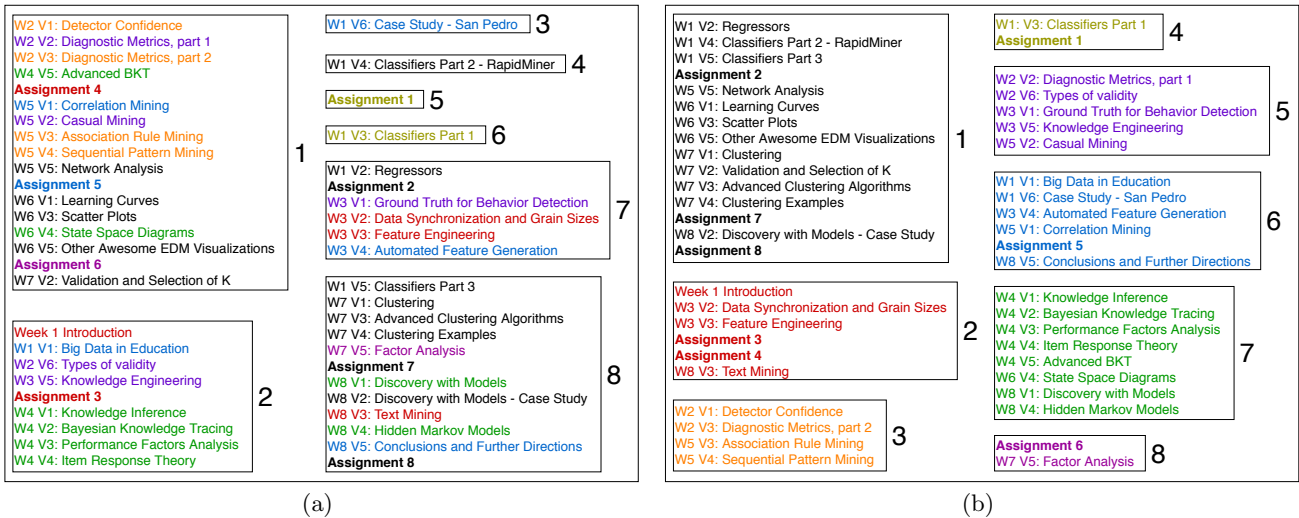


Figure 7: Clusters discovered by using MVKM (a), clusters discovered by using video-lecture transcripts and assignment texts(b).

similarities in video lectures that are clustered together by MVKM. For example, videos 1 and 3 of week 2 are clustered with later weeks' videos because of content similarity (cluster 1 in Figure 7(a)). While videos 2 of week 2 is also clustered with them because it comes between these two videos in course sequence.

Additionally, between video lectures and assignments, the clusters closely follow the course structure. The assignments in this course come at the end of their module and right before the next module starts. For example, "Assignment 3" appears after video 5 at week 3 and before video 1 at week 4. We can see that all assignments, except "Assignment 1" that is the first one, are clustered with their immediate next video lecture. Moreover, we can see the effect of content similarity between assignments and video lectures in differences of Figures 6(a) and 7(a). For example, without including assignments, "Week 1 Introduction" and "W1 V1: Big Data in Education" were clustered together in cluster 7 of Figure 6(a). However, after adding assignments, because of the content similarity between "Assignment 3" and "Week 1 Introduction" (Figure 7(b) cluster 2), "Week 1 Introduction" and "W1 V1: Big Data in Education" are clustered with video lectures that are structurally close to "Assignment 3".

Altogether, we demonstrated that learning materials' bias parameters in MVKM are aligned with their difficulties; learning materials' latent concepts discovered by our model well represent learning materials' real-world similarities, both in structure and in content; and MVKM can successfully unveil these similarities between different types of learning materials, without observing their content or structure.

5. CONCLUSIONS

In this paper, we proposed a novel Multi-View Knowledge Model (MVKM) that can model students' knowledge gain from different learning materials types, while simultaneously discovering materials' latent concepts. Our proposed tensor factorization model explicitly represents students' knowledge growth and allows for occasional forgetting of learned concepts. Our extensive evaluations on synthetic and real-world datasets show that MVKM outperforms other baselines in the task of student performance prediction, can effectively capture students' knowledge growth, and represent similarities between different learning materials types.

6. ACKNOWLEDGMENTS

This paper is based upon work supported by the National Science Foundation under Grant No. 1755910.

7. REFERENCES

- [1] R. Agrawal, M. Christoforaki, S. Gollapudi, A. Kannan, K. Kenthapadi, and A. Swaminathan. Mining videos from the web for electronic textbooks. In *International Conference on Formal Concept Analysis*, pages 219–234. Springer, 2014.
- [2] R. Agrawal, S. Gollapudi, A. Kannan, and K. Kenthapadi. Enriching textbooks with images. In *Proceedings of the 20th ACM International Conference on Information and knowledge management*, pages 1847–1856, 2011.
- [3] G. Alexandron, Q. Zhou, and D. Pritchard. Discovering the pedagogical resources that assist students in answering questions correctly—a machine learning approach. In *the 8th International Conference on Educational Data Mining (EDM 2015)*, pages 520–523, 2015.
- [4] J. M. L. Andres, R. S. Baker, G. Siemens, D. Gašević, and C. A. Spann. Replicating 21 findings on student success in online learning. *Technology, Instruction, Cognition, and Learning*, pages 313–333, 2016.
- [5] R. S. Baker. *Big Data and Education*. New York, NY: Teachers College, Columbia University, 2nd edition, 2015.
- [6] R. S. Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *International conference on intelligent tutoring systems*, pages 406–415. Springer, 2008.
- [7] T. Barnes. The q-matrix method: Mining student response data for knowledge. In *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, pages 1–8, 2005.
- [8] J. E. Beck, K.-m. Chang, J. Mostow, and A. Corbett. Does help help? introducing the bayesian evaluation and assessment methodology. In *International Conference on Intelligent Tutoring Systems*, pages 383–394. Springer, 2008.
- [9] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [10] A. F. Botelho, S. Adjei, and N. T. Heffernan. Modeling interactions across skills: A method to construct and compare models predicting the existence of skill relationships. In *EDM*, pages 292–297, 2016.
- [11] Canvas-Network. Canvas network courses, activities, and users (4/2014 - 9/2015) restricted dataset, 2016.
- [12] G. Casalino, C. Castiello, N. Del Buono, F. Esposito, and C. Mencar. Q-matrix extraction from real response data using nonnegative matrix factorizations. In *International Conference on Computational Science and Its Applications*, pages 203–216. Springer, 2017.
- [13] Y. Chen, J. P. González-Brenes, and J. Tian. Joint discovery of skill prerequisite graphs and student models. In *The 9th International Educational Data Mining*, pages 46–53, 2016.
- [14] B. Choffin, F. Popineau, Y. Bourda, and J.-J. Vie. Das3h: Modeling student learning and forgetting for optimally scheduling distributed practice of skills. In *The 12th International Conference on Educational Data Mining*, pages 384–389. IEDMS, 2019.
- [15] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.
- [16] M. Desmarais, P. Xu, and B. Beheshti. Combining techniques to refine item to skills q-matrices with a partition tree. In O. Santos, J. G. Boticario, C. Romero, M. Pechenizkiy, A. Merceron, P. Mitros, J. M. Luna, C. Mihaescu, P. Moreno, A. Hershkovitz, S. Ventura, and M. Desmarais, editors, *the 8th International Conference on Educational Data Mining (EDM 2015)*, pages 29–36, 2015.
- [17] T.-N. Doan and S. Sahebi. Rank-based tensor factorization for student performance prediction. In *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, volume 288, page 293. ERIC.
- [18] T.-N. Doan and S. Sahebi. Rank-based tensor factorization for predicting student performance. In *The 12th International Conference on Educational Data Mining*, pages 288–293. IEDMS, 2019.
- [19] F. Drasgow and C. L. Hulin. Item response theory. *Handbook of industrial and organizational psychology*, 1:577–636, 1990.
- [20] J. González-Brenes. Modeling skill acquisition over time with sequence and topic modeling. In *Artificial Intelligence and Statistics*, pages 296–305, 2015.
- [21] J. P. González-Brenes, Y. Huang, and P. Brusilovsky. Fast: Feature-aware student knowledge tracing. In *NIPS Workshop on Data Driven Education*, 2013.
- [22] R. Hosseini, T. Sirkä, J. Guerra, P. Brusilovsky, and L. Malmi. Animated examples as practice content in a java programming course. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, SIGCSE ’16, pages 540–545, New York, NY, USA, 2016. ACM.
- [23] Y. Huang, J. P. González-Brenes, and P. Brusilovsky. Challenges of using observational data to determine the importance of example usage. In *International Conference on Artificial Intelligence in Education*, pages 633–637. Springer, 2015.
- [24] M. J. Johnson. Scaling cognitive modeling to massive open environments.
- [25] M. M. Khajah, Y. Huang, J. P. González-Brenes, M. C. Mozer, and P. Brusilovsky. Integrating knowledge tracing and item response theory: A tale of two frameworks. In *Proceedings of Workshop on Personalization Approaches in Learning Environments (PALE 2014) at the 22th International Conference on User Modeling, Adaptation, and Personalization*, pages 7–12. University of Pittsburgh, 2014.
- [26] K. R. Koedinger, J. C. Stamper, E. A. McLaughlin, and T. Nixon. Using data-driven discovery of better student models to improve student learning. In *Artificial intelligence in education*, pages 421–430. Springer, 2013.
- [27] A. S. Lan, C. Studer, and R. G. Baraniuk. Time-varying learning and content analytics via sparse factor analysis. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 452–461, 2014.
- [28] A. S. Lan, C. Studer, and R. G. Baraniuk.

- Time-varying learning and content analytics via sparse factor analysis. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 452–461. ACM, 2014.
- [29] R. V. Lindsey, J. D. Shroyer, H. Pashler, and M. C. Mozer. Improving students’ long-term knowledge retention through personalized review. *Psychological science*, 25(3):639–647, 2014.
- [30] R. Liu, J. L. Davenport, and J. C. Stamper. Beyond log files: Using multi-modal data streams towards data-driven kc model improvement. In *The 9th International Educational Data Mining*, pages 436–441, 2016.
- [31] M. C. Mozer and R. V. Lindsey. Predicting and improving memory retention: Psychological theory matters in the big data era. In *Big data in cognitive science*, pages 43–73. Psychology Press, 2016.
- [32] K. Nagatani, Q. Zhang, M. Sato, Y.-Y. Chen, F. Chen, and T. Ohkuma. Augmenting knowledge tracing by considering forgetting behavior. In *The World Wide Web Conference*, pages 3101–3107. ACM, 2019.
- [33] A. S. Najar, A. Mitrovic, and B. M. McLaren. Adaptive support versus alternating worked examples and tutored problems: Which leads to better learning? In *International Conference on User Modeling, Adaptation, and Personalization*, pages 171–182. Springer, 2014.
- [34] Z. Pardos, Y. Bergner, D. Seaton, and D. Pritchard. Adapting bayesian knowledge tracing to a massive open online course in edx. In *Educational Data Mining 2013*, 2013.
- [35] Z. A. Pardos and N. T. Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 255–266. Springer, 2010.
- [36] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis: a new alternative to knowledge tracing. In *14th International Conference on Artificial Intelligence in Education*, volume 2009, pages 531–538, 2009.
- [37] R. Pelánek. Measuring similarity of educational items: An overview. *To appear in IEEE Transactions on Learning Technologies*, 2019.
- [38] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in neural information processing systems*, pages 505–513, 2015.
- [39] Y. Qiu, Y. Qi, H. Lu, Z. A. Pardos, and N. T. Heffernan. Does time matter? modeling the effect of time with bayesian knowledge tracing. In *EDM*, pages 139–148, 2011.
- [40] S. Sahebi. *Canonical Correlation Analysis in Cross-Domain Recommender Systems*. PhD thesis, Intelligent Systems Program, University of Pittsburgh, 2016.
- [41] S. Sahebi and P. Brusilovsky. Student performance prediction by discovering inter-activity relations. *Educational Data Mining*, pages 87–96, 2018.
- [42] S. Sahebi, Y. Lin, and P. Brusilovsky. Tensor factorization for student modeling and performance prediction in unstructured domain. In *The 9th International Conference on Educational Data Mining*, pages 502–506. IEDMS, 2016.
- [43] M. Sao Pedro, R. Baker, and J. Gobert. Incorporating scaffolding and tutor context into bayesian knowledge tracing to predict inquiry skill acquisition. In *Educational Data Mining 2013*, 2013.
- [44] N. Thai-Nghe, T. Horvath, and L. Schmidt-Thieme. Context-aware factorization for personalized student’s task recommendation. In *Proceedings of the International Workshop on Personalization Approaches in Learning Environments*, volume 732, pages 13–18, 2011.
- [45] W. J. van der Linden and R. K. Hambleton. *Handbook of modern item response theory*. Springer Science & Business Media, 2013.
- [46] N. F. Velasquez, I. Goldin, T. Martin, and J. Maughan. Learning aid use patterns and their impact on exam performance in online developmental mathematics. In *Educational Data Mining 2014*, 2014.
- [47] J.-J. Vie and H. Kashima. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 750–757, 2019.
- [48] L. Wang, A. Sy, L. Liu, and C. Piech. Learning to represent student knowledge on programming exercises using deep learning. In *Proceedings of the 10th International Conference on Educational Data Mining*, pages 324–329, 2017.
- [49] M. Wen and C. P. Rosé. Identifying latent study habits by mining learner behavior patterns in massive open online courses. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1983–1986. ACM, 2014.
- [50] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SIAM International Conference on Data Mining*, volume 10, pages 211–222, 2010.
- [51] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *International Conference on Artificial Intelligence in Education*, pages 171–180. Springer, 2013.
- [52] M. V. Yudelson, O. P. Medvedeva, and R. S. Crowley. A multifactor approach to student model evaluation. *User Modeling and User-Adapted Interaction*, 18(4):349–382, 2008.
- [53] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pages 765–774. International World Wide Web Conferences Steering Committee, 2017.

Predicting Student Performance in a Master of Data Science Program using Admissions Data

Yijun Zhao
Computer and Information
Science Department
Fordham University
New York, NY
yzhao11@fordham.edu

Qiangwen Xu
Computer and Information
Science Department
Fordham University
New York, NY
qxu47@fordham.edu

Ming Chen
Computer and Information
Science Department
Fordham University
New York, NY
mchen177@fordham.edu

Gary M. Weiss
Computer and Information
Science Department
Fordham University
New York, NY
gaweiss@fordham.edu

ABSTRACT

Predicting student success in a data science degree program is a challenging task due to the interdisciplinary nature of the field, the diverse backgrounds of the students, and an incomplete understanding of the precise skills that are most critical to success. In this study, the applicant's future academic performance in a Master of Data Science program is assessed using information from the admission application, such as standardized test scores, undergraduate grade point average, declared major, and school ranking. Simple data analysis methods and visualization techniques are used to gain a better understanding of how these variables impact student performance, and several classification algorithms are used to induce models to distinguish between students that will perform very well and those that will perform very poorly. Historical admissions and grading data are used to perform these analyses and build the classification models. The analyses and predictive models that are generated provide insight into the factors that identify good and poor candidates, and can aid in future admissions decisions.

Keywords

Admission decision making, Master's program, data science, learning assessment, machine learning.

1. INTRODUCTION

Data mining methods are now in widespread use in many industries, from healthcare[10] to business[15]. Data mining is increasingly applied to education [3][8][14] and includes many diverse applications, all of which fall under the area of educational data mining (EDM). A particular focus of

such applications is the college admissions process and its effectiveness, since this process directly affects the reputation of the institution as well as its financial well-being. Examples of work in this area include predicting college admissions yield [5], student retention[11], and enrollment management[1]. Another related area of EDM relates to predicting student performance. One such study used student personal and social factors, along with academic performance data, to identifying poor performers early on[2], while another study used similar information to predict third semester academic performance [13]. One more study used student course data during the semester (attendance, homework scores, etc.) to predict the student score on the end of the semester examination[18].

In this paper we investigate the problem of identifying a good admissions strategy for a Master's of Science program in Data Science (MSDS), so that the students that are admitted into the program will perform well. This problem is generally related to the EDM admissions topic, but also to the topic of predicting student performance. This problem is interesting, and distinctive, for a variety of reasons. One reason is that the vast majority of applications of data mining to college admissions deals with undergraduate admissions. That admissions process is very different from the process for our MSDS program, since undergraduate admissions is controlled by full-time admissions professionals, whereas admissions for our MSDS program is controlled by faculty with little time to devote to admissions, and who lack specialized admissions training. This is true for most graduate programs, except for possibly the large professional schools (e.g., law, medicine) that may admit many more students and have deeper resources. Determining admission to MSDS programs is especially challenging since it is an interdisciplinary field that attracts applicants from diverse backgrounds, and because MSDS programs were introduced only recently and hence have limited historical knowledge to leverage. Furthermore, even experts in the area do not fully understand exactly which undergraduate skills are most critical to success, so it is hard to know which students to admit or reject.

Yijun Zhao, Qiangwen Xu, Ming Chen and Gary Weiss "Predicting Student Performance in a Master's Program in Data Science using Admissions Data" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 325 - 333

The goal of this data mining task is to determine if, using only information available in the admissions application, a student will perform very well in the program, and hence deserve merit-based aid, or perform very poorly and hence should not be admitted. At this time only structured data is utilized in order to simplify the classification task. Thus external recommendations, personal statements, and resumes are not considered. However, there is still a wealth of information that is available, which includes prior degrees and associated grades, the name and country of the prior educational institutions, standardized test scores such as the GRE (Graduate Record Examination) and TOEFL (Test of English as a Foreign Language), and personal information about the applicant such as age, nationality, work history, and whether they ask for merit-based financial aid.

The purpose of this study is not just to identify which students will perform very well or very poorly, but to better understand the relevant factors. Thus, the predictive model that we build will most likely not be used for automated decision making, but instead will be used to educate the admissions committee about which factors are most relevant for success in the program. As mentioned earlier, this is especially important for the MSDS degree because the applicants have such different backgrounds and because the degree is relatively new.

A practical issue that impacts this study is that because the offered degree was launched only a few years ago, the data is quite limited. Compounding this issue is the fact that we do not have outcomes for students who are accepted but do not attend the university, and worse yet, we cannot know anything about how students who are rejected from the program would perform. One of our long term goals is to fully utilize this unlabeled data to improve the admissions process. This is discussed later in this paper as future work.

The rest of the paper is organized as follows. We present the details of our dataset in Section 2. The design of our experiments and associated methodology are presented in Section 3. Section 4 presents our experimental results and predictive factor analysis. We conclude and suggest future work in Section 5.

2. THE DATA

This section describes the data utilized in this study. Section 2.1 describes the data at a high level and includes some summary statistics, while Section 2.2 describes the features included in each application record. Section 2.3 then describes the distribution of feature values for key features, while Section 2.4 describes how these feature values relate to student performance in the MSDS program.

2.1 Overview

The data in this study is extracted from the application data provided by each applicant to Fordham University's MSDS program. The application process is completely electronic, so the underlying data was already in electronic form. Much of this data is structured data (e.g., student GRE scores) that is already parsed and readily available for extraction. However, other information, such as the student's required statement of purpose and external recommendations, are un-

structured text and would require substantial effort to exploit. This study is limited to structured data.

The data set is comprised of 826 applicant records. Of this total, 503 (60.9%) applicants were accepted into the program and 323 (39.1%) were rejected. Of the 503 accepted applicants, 132(26.2%) enrolled in the program while 371 (73.8%) did not enroll. Since only students who enrolled have grade information, the main analyses presented in this paper are based on only 132 records. Note that the data used in these analyses depends on our current admissions strategy, since it is possible that some of the students who were denied admission into the program could have performed well in the classes. The best we can do with respect to the population of students who were denied admission is to compare their characteristics with those of the students who enrolled and performed poorly; if it turns out that the factors used to determine admission into the program differ from those that tend to predict good performance, then the current admissions strategy should be modified. The characteristics of each student population are explored and compared in Section 2.3.

2.2 Features and Feature Generation

The features that are extracted from the student applications and used in this study are listed in Table 1, along with sample values. The first three features describe the Graduate Record Examination (GRE) standardized test scores and are encoded using the score percentile. The fourth feature describes the Test of English as a Foreign Language (TOEFL) total score. The next field specifies the number of months from the time of completion (or projected completion) of the last degree to the time the current application was submitted. For students who plan to start the MSDS program immediately following the graduation from their current program, this value is typically a negative six months. Student age is at the time of application and marital status is single, married, divorced, domestic partner, or blank (unspecified). Gender is either male or female and citizenship specifies the country of citizenship. The next six features relate to the last degree program (i.e., school) that the student attended. They include the student's GPA (Grade Point Average), major and degree, the country that the school resides in, the primary language of instruction, and the school ranking. The MSDS GPA uses a 4-point scale, and is based on the student's performance after enrolling in the program. This attribute is utilized to generate the class value, as described in Section 3.1.

School rank is the only feature in Table 1 that is not a feature from the student application. Instead, the ranking is generated from the school name via a multi-step process. The first step involves matching the school name against the US News and World Report "Best Global Universities" ranking ([usnews.com/education/best-global-universities](https://www.usnews.com/education/best-global-universities)), which includes 1500 universities from eighty countries and is based on academic research performance and global and regional reputation. If a match is found, then this global ranking is used; otherwise the US News and World Report "Best Colleges" ranking is searched ([usnews.com/best-colleges](https://www.usnews.com/best-colleges)). This is restricted to colleges in the United States and includes separate rankings for national universities (major research institutions), liberal arts colleges, and regional

Table 1: Data Set Features and Sample Values

#	Feature Name	Sample Value
1	GRE Verbal %	52
2	GRE Quantitative %	95
3	GRE Writing %	34
4	TOEFL Total	105
5	Months since Degree	6
6	Student Age	22
7	Marital Status	Married
8	Gender	M
9	Citizenship	China
10	School GPA	3.7
11	School Major	Chemical Engineering
12	School Degree	BS
13	School Country	China
14	School Language	Mandarin
15	School Rank	85
16	MSDS GPA	3.5

colleges and universities. If a match is found in one of these rankings, then the ranking is converted to a global ranking by adding 1200 if the match was for a national ranking, and 1400 if the match was for a regional ranking. This process of assigning a global ranking is a very rough heuristic method, but generally provides reasonable values. If a school is not found on any of these rankings then a global ranking of 9999 is used.

There are a number of features that are available from the original application information but are not used in this study and do not appear in Table 1. For example, the Total TOEFL score is included but the four TOEFL subscores are not included, since preliminary analysis indicated that these subscores did not provide much benefit. Additionally, if the applicant attended multiple institutions of higher education, then information for more than one school was provided. However, since providing additional schools for only some applicants would substantially complicate the analysis, this information was dropped, so only the most recent degree granting school was included.

2.3 Distribution of Feature Values

In any applied data mining study, it is important to understand the data. In this section, we provide information about the distribution of feature values. Since the focus of this study is in identifying students who will perform well or poorly in the program, we begin with the feature distribution of the students who enrolled in the MSDS program. This information is provided in Figure 2.2. The figure provides a good overview of the demographics of the applicants: males outnumber females by a ratio of almost 2 to 1, nearly 90% are single, and based on citizenship, about 74% are foreign nationals, while 21% are US citizens, and 5% are permanent residents. Clearly the MSDS program attracts a large international contingent. As expected, most students are young, although about 6% are over 30, suggesting that they likely have substantial industry experience. Overall, more than 80% are within two years of their last degree.

The TOEFL scores, which are only required for international students who have not completed two years of instruction

at an English-language university, show that most students, but not all, have good English language skills. According to the testing agency, the average TOEFL score is 84, and any such score is generally considered good. For the MSDS program, a score of 80 or above is generally required, and hence our admitted students tend to have good English language skills.

A feature that is critical to the admissions decision is the student's prior major discipline. The program is geared towards students who have substantial mathematics background and at least some experience in computer science and programming. While computer science and mathematics majors are thought to have an advantage, students in any science or quantitative discipline are encouraged to apply. Figure 2.2 shows the distribution of major over all applicants, and further shows the number in each major that were admitted and rejected. The statistics show that the largest number of admitted students have a background in computer science or a highly related field, with mathematics and statistics a close second.

Although this study focuses on enrolled students, it is useful to understand the characteristics of the students who were rejected from the program, or were admitted and did not enroll, and how they compare to students who did enroll. Displaying this information graphically for all three populations would take up too much space, so the key observations are summarized below.

- Applicants who were rejected are much more likely to have a GPA under 3.0 (41%) than those who enrolled (14%) or were admitted and did not enroll (17%).
- Applicants who were rejected are more likely to have a degree from an institution not ranked in the top 2000 (30.7%) than those who enrolled (25.8%) or were admitted but did not enroll (20.2%).
- Applicants who enrolled were about 6% more likely to have completed a graduate degree (21.2%) than those who were admitted but did not enroll (15.1%) or were rejected (14.6%).
- Applicants who enrolled were less likely to be female (35.6%) than those who were admitted but did not enroll (47.4%) or were rejected (39.6%). Female applicants who are admitted are less likely to enroll than their male counterparts.
- Foreign nationals made up 74% of enrolled students, 79% of those admitted who did not enroll, and 77% of those rejected. As might be expected, foreign nationals who are admitted are somewhat less likely to attend.
- The age profile does not vary much between those applicants who enroll and are rejected. However, applicants who enroll are much more likely to be older and between the ages of 24 and 30 (33%) than those who are admitted but do not enroll (22%).
- Those who apply more than two years after completing their last degree are more likely to be rejected from the program (33.3%) than those who either enroll (17%) or are admitted and do not enroll (19%).

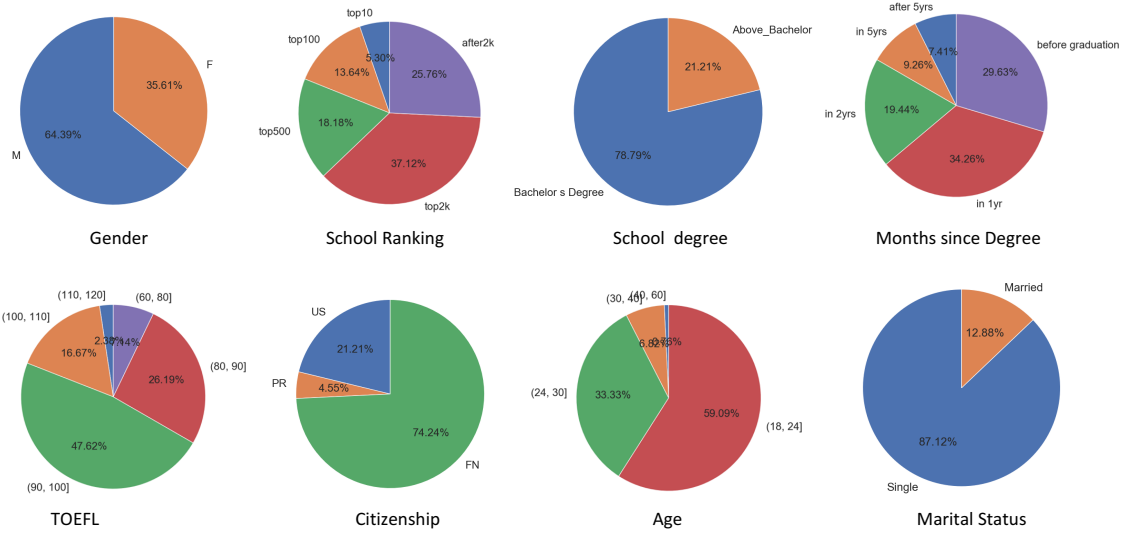


Figure 1: Feature Statistics

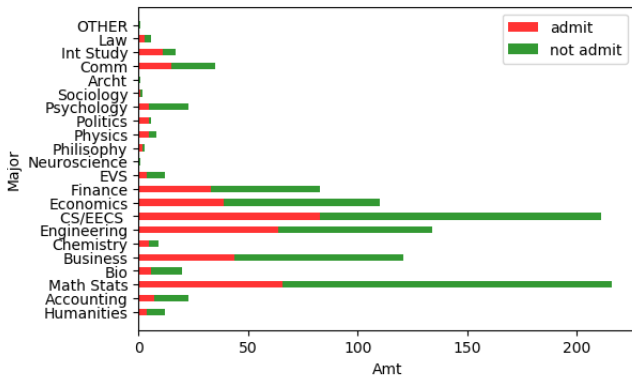


Figure 2: Major Distribution

Table 2: Mean Values for MSDS Performance Groups

Feature	Bottom 20%	Middle 60%	Top 20%	All
GRE Verbal %	42.5	48.6	57.0	49.4
GRE Quantitative %	<u>79.0</u>	81.9	<u>82.8</u>	81.6
GRE Writing %	32.5	31.2	34.0	32.0
TOEFL Total %	96.1	96.1	96.6	96.2
Foreign National %	<u>59.1</u>	74.4	<u>85.7</u>	74.2
Graduate Degree %	<u>18.2</u>	19.5	<u>28.6</u>	21.2
Married %	22.7	9.8	14.3	12.9
Female %	22.7	39.0	35.7	35.6
Months Since Degree	<u>12.8</u>	13.6	<u>22.8</u>	15.4
Age	25.5	24.7	24.6	24.8
School Rank	1005	1005	1082	1005
School GPA	<u>3.17</u>	3.29	<u>3.53</u>	3.32

2.4 Features and MSDS Performance Groups

In Section 2.3 the distribution of feature values was examined for the population of enrolled students, and then the key differences between the features for the three admission categories (enrolled, admit but not enrolled, rejected) was analyzed. In this section, we compare the feature values with respect to student performance in the MSDS program to provide insight into the factors that influence student performance. As will be discussed in Section 3.1, our focus in this study is to identify the students who enroll in the MSDS program that will perform in the top 20% and the bottom 20%. Thus, in this section, we examine the feature values for three performance groups: the bottom 20%, middle 60%, and top 20%. In order to simplify the comparison, the mean values of numerical features are considered. Table 2 provides the relevant information. The features values that differ substantially between the bottom and top 20%, and we believe are of predictive value, are underlined.

The three GRE test score percentiles in Table 2 show the expected trend: the scores improve as we move up the performance groups. The one exception is that there is a slight dip in the GRE writing score when moving from the bottom group to the middle group, but even in this case the writing scores for the top group outperform the bottom group. Our admissions committee normally places the most weight on the quantitative score and generally does not consider the writing score. What is most interesting is that the difference in the quantitative scores between the bottom and top 20% is only 3.8% (79.0% vs. 82.8%), even though quantitative abilities are generally thought to be critical for data scientists. The modest difference may reflect the fact that the GRE exam only tests fundamental mathematical skills. The TOEFL score barely differs between the three performance groups; however, this may not be surprising since the mean values are quite high, indicating that most students have more than sufficient English language skills.

There is an obvious pattern with respect to the percentage of foreign nationals—the percentage of foreign nationals increases from 59.1% for the bottom 20% to 85.7% for the top 20%. This marked difference occurs even though we showed in Section 2.3 that the percentage of foreign nationals is relatively constant across the three admissions categories. It is worth mentioning that many of the foreign nationals completed their undergraduate education in the United States. The data also shows that a higher percentage of students in the top 20% previously earned a graduate degree. This may seem intuitive, but since these degrees are generally in very different disciplines, the relationship is not obvious. Such students will have demonstrated the prior ability to complete graduate work and perhaps the maturity associated with this is a significant factor. A graduate degree is considered quite favorably in the admissions process, and also enables a student to compensate for a low undergraduate GPA.

Students in the top 20% are less likely to be married than those in the bottom 20%, but the trend is not consistent through the middle 60%, so we tend not to place too much weight on these differences. The students in the top 20% are more likely to be female than those in the bottom 20% (35.7% versus 22.7%) and this suggests that there is a real gender difference. The explanation for these gender differences is not obvious, but at the undergraduate level we have observed that academically weak female students tend not to major in Computer Science due to the societal pressure that already discourages them from majoring in scientific and technical disciplines.

The next two features show that higher performing students tend to have more time since the granting of their last degree (about one more year), but are still approximately the same age. This could reflect the fact that students who immediately proceed from an undergraduate degree to the MSDS program have not thought as deeply about their desire to become a data scientist and hence may not be as committed. The school rank does not differ significantly between the performance categories, suggesting that the reputation of the prior school is not a key factor in student performance in the MSDS program. Finally, there is a very clear trend that the higher the GPA in the prior degree, the higher performing the student. This is perhaps the most obvious indicator of future achievement and the values support that: the prior GPA of those that are in the top 20% of the MSDS program is 3.53 versus 3.17 for those in the bottom 20%.

3. EXPERIMENT METHODOLOGY

This section describes the experiments related to predicting student academic performance in the MSDS program. Section 3.1 precisely defines the problem as a classification problem. Section 3.2 provides a brief description of the eight classification algorithms utilized in this study. The details concerning the design of the experiments are provided in Section 3.3.

3.1 Problem Formulation

We are primarily interested in identifying the applicants that will perform very well and will have GPAs within the top 20% of enrolled MSDS students, or will perform poorly and fall within the bottom 20%. The reason for this is that

we want to deny admission to those who we anticipate will perform in the bottom 20% and may want to provide merit-based aid to those we expect to perform in the top 20%. Note that this does not mean we only deny admittance to the bottom 20%, since admission will already be denied to those who do not meet our general admissions requirements (e.g., GPA above 3.0, TOEFL above 80, etc.). We therefore build two classification models: one that distinguishes the top 20% from the bottom 80% and one that distinguishes the bottom 20% from the top 80%. The minority class is always considered the positive class. The performance of these two models is described in Section 4.

3.2 Classification Algorithms

This section provides brief descriptions of the established machine learning algorithms that are employed in this study. A heterogeneous ensemble approach is also described.

3.2.1 Logistic Regression

Logistic Regression [12] is a type of generalized linear model (GLM) that studies the association between a categorical response variable Y and a set of independent (explanatory) variables $X = \{X_1, X_2, \dots, X_n\}$. In particular, the Y variable is first modeled as a linear function of X , and then the numerical predictions of Y are transformed into probability scores using a sigmoid function. In a binary classification task, the scores indicate a corresponding instance's likelihood of belonging to the positive class. Thus, a cutoff (usually 0.5) can be established as a decision boundary to further categorize the instances into the more likely class.

3.2.2 Support Vector Machines (SVM)

SVM [6] performs classification tasks by constructing a decision boundary in a multidimensional space that separates instances of different class labels. *SVM* strives to maximize the distance between the hyperplane and the data points of both classes. Maximizing the margin distance reinforces that future data points can be classified with more confidence. *SVM* is capable of transforming the data into a higher dimensional space using various kernel functions to enhance data separability. In this study linear *SVM* is used to facilitate risk factor analysis.

3.2.3 Decision Trees

A *Decision Tree* [16] model uses a tree structure to model the data in which each leaf node corresponds to a class label and attributes are represented as the internal nodes of the tree. Each branch represents a potential value of its parent node (i.e., an attribute). The major challenge in building a *Decision Tree* model is to choose the attribute for each node in each level. In our study we use the *Gini Index* as our criterion for attribute selection.

3.2.4 Random Forest

Random Forest [4] is a collection of decision trees, where each tree is trained with a subset of training instances and a subset of attributes. By pooling predictions from multiple decision trees, *Random Forest* reduces the variance of each individual tree and achieves a more robust and superior performance.

3.2.5 Neural Network

A *Neural Network* [9] is a computational model that is inspired by the way biological neural networks in the human brain process information. It consists of an input layer, one or more hidden layer(s), and one output layer. The adjacent layers are connected by transferring the values in one layer to a new set of values in the next layer with a set of weights and an activation function. “Training” is the process of adjusting the network weights using a back propagation algorithm to achieve the highest consistency (i.e., cross entropy) between the model outputs and the true class labels.

3.2.6 Naive Bayes

A *Bayes classifier* belongs to the family of probabilistic generative models. The algorithm differs from discriminative models in that, instead of finding a functional form, it models the probability distributions of the data. In a binary classification task, predictions are set to the larger of $P(y = i|X)$ where $i \in \{0, 1\}$ and $X = \{x_1, x_2, \dots, x_d\}$. A *Naive Bayes* classifier further assumes that features are independent of each other given the class, which simplifies the evaluation of $P(X|y=i)$ to $\prod_{j=1}^d p(x_j|y=i)$.

3.2.7 K-Nearest Neighbor (KNN)

KNN is an effective classification algorithm that does not require pre-training of a model. Classification decisions are based on a majority vote on k empirically observed instances that are most similar to the instance in question. The resemblance is typically measured by a distance metric such as Euclidean distance operated on the attributes describing the two instances.

3.2.8 Ensemble Learner L

In addition to individual machine learning algorithms, we explored ensemble techniques [7] to integrate information from different classifiers. Ensemble learning is a family of algorithms that seek to create a “strong” classifier based on a group of “weak” classifiers. In this context, “strong” and “weak” refer to how accurately the classifiers can predict the target variable. Ensemble learning has been proven to have improved and more robust performance than a single model. Specifically, multiple base classifiers are built for the original classification task with the training data. A meta-learner L is constructed by combining the outcomes from the base classifiers to improve predictive accuracy. In this study we combine the predictions from the base classifiers using an unweighted majority vote and our base learners consists of seven single models described in Sections 3.2.1 - 3.2.7.

3.3 Experiment Design

All experiments in this study utilize 10-fold cross validation. In addition to reporting overall predictive accuracy, the results in Section 4 and Table 3 report the performance on the positive/minority class via the sensitivity metric, which is also known as recall and true positive rate, and the performance on the majority/negative class via the specificity metric, which is also known as true negative rate. For both classification tasks, there is class imbalance since the ratio of the positive to negative class is approximately 1:4. Bagging is used to address this class imbalance; at training time five bags of balanced training data are created where each bag

consists of all minority-class examples and an equal number of randomly selected majority-class examples. The class for each test example is based on a majority vote of the five models built using the data from each bag.

The parameters of the models are selected experimentally using the training data using a grid search. Both the training and test accuracies are reported in Table 3. Specifically, for the SVM model, the trade-off parameter $C = 0.1$. For the KNN algorithms, the number of nearest neighbors $k = 3$. For the neural networks model, we used a 3-layer architecture with (128, 256, 512) nodes in identifying the bottom 20% of the students, and a two-layer architecture with (128, 256) nodes in identifying the top 20% of the students. For the rest of the algorithms, including the depth of the decision tree, the number of trees in the random forest, etc., we applied the default parameters provided by the Python scikit-learn package.

4. EXPERIMENT RESULTS

This section presents the results of the classification experiments. The accuracy results for identifying the top and bottom performing students are presented, as are the top predictors for identifying these two populations.

4.1 Analysis on Performance Measures

Table 3 presents all of the performance results for the two classification tasks. This analysis focuses exclusively on the performance on the test data. The results in the table show that *Random Forest* and the ensemble learner L achieve the two best overall predictive accuracy values for both classification tasks. For the tasking of identifying the bottom 20% of students, L achieved an 86% overall accuracy compared to 83% for random *Random Forest*. When these results are broken down into performance on the bottom 20% and the rest, L achieved results of 90% and 83%, respectively, versus 91% and 75% for *Random Forest*. It should be noted, however, that although *Decision Tree* has only the third best overall performance, it has the best performance at identifying the bottom 20% of the students (94% versus 91% for *Random Forest* and 90% for L). However, *Decision Tree* performs very poorly at classifying the remaining 80%, with a specificity of 65%.

For the classification task of identifying the top 20% of students, *Random Forest* delivered an overall accuracy of 86%, while L achieved an overall accuracy of 85%. When these results are broken down into performance on the top 20% and the rest, *Random Forest* achieved results of 94% and 79%, respectively, versus 92% and 79% for L . In this case *Decision Tree* again did very well when just evaluated on the minority class, with a performance of 94% for the top 20%, equalling the performance of *Random Forest* on this population. Note that since *Random Forest* is a collection of decision trees, it belongs to the family of *homogeneous* ensemble methods. Thus, we conclude that ensemble learners are the best machine learning models for the two classification tasks.

4.2 Analysis on Predictive Features

An additional motivation of our research is to identify the top predictors for the successful and struggling students.

Table 3: Performance Comparison Over Eight Models

Models	Bottom 20% vs. Rest						Top 20% vs. Rest					
	Test			Training			Test			Training		
	Bot20	Rest	Overall	Bot20	Rest	Overall	Top20	Rest	Overall	Bot20	Rest	Overall
SVM	0.74	0.62	0.68	0.85	0.69	0.77	0.68	0.54	0.61	0.75	0.58	0.67
Decision Tree	0.94	0.65	0.80	0.95	0.80	0.87	0.94	0.67	0.80	0.96	0.80	0.88
Random Forest	0.91	0.75	0.83	0.96	0.94	0.95	0.94	0.79	0.86	0.96	0.85	0.90
Logistic Regression	0.71	0.63	0.67	0.84	0.74	0.79	0.71	0.63	0.67	0.86	0.75	0.81
KNN	0.93	0.63	0.78	1.00	0.82	0.91	0.90	0.58	0.73	0.98	0.70	0.84
Naive Bayes	0.83	0.54	0.68	0.91	0.58	0.74	0.58	0.58	0.58	0.72	0.67	0.70
Neural Network	0.34	0.80	0.57	0.39	0.83	0.61	0.52	0.53	0.53	0.59	0.57	0.58
Ensemble (L)	0.90	0.83	0.86	0.96	0.92	0.94	0.92	0.79	0.85	0.96	0.87	0.91

Table 4: List of Top 10 Predictive Features in Identifying Bottom 20% of Students

Predictors of the Bottom 20% vs. Rest Models				
Rank	SVM	Logistic Regression	Random Forest	Decision Tree
1	Economics ¹	Economics ¹	GRE Verbal %	GRE Verbal %
2	China ²	Environmental Studies ¹	Months since Degree	Months since Degree
3	Communications ¹	US ³	GRE Quantitative %	GRE Writing %
4	Environmental Studies ¹	CS/EECS ¹	GRE Writing %	Economics ¹
5	Psychology ¹	Business ¹	School Rank	GRE Quantitative %
6	CS/EECS ¹	Communications ¹	Student Age	School Rank
7	Applied Math/Stats ¹	FN ³	Overall GPA	Business ¹
8	Masters ⁴	Biochemistry/Biology ¹	Psychology ¹	FN ³
9	FN ³	Bachelors ⁴	Economics ¹	Overall GPA
10	Bachelors ⁴	Architecture ¹	FN ³	Student Age
Predictors of the Top 20% vs. Rest Models				
Rank	SVM	Logistic Regression	Random Forest	Decision Tree
1	Business ¹	Business ¹	Overall GPA	Months since Degree
2	Engineering ¹	International Studies ¹	GRE Verbal %	GRE Verbal %
3	Overall GPA	Bachelors ⁴	GRE Writing %	Overall GPA
4	CS/EECS ¹	US ³	Student Age	Student Age
5	Bachelors ⁴	Chemistry ¹	GRE Quantitative %	GRE Quantitative %
6	International Studies ¹	Humanities ¹	School Rank	School Rank
7	China ²	Accounting ¹	Months since Degree	TOEFL Total
8	Accounting ¹	Finance ¹	Business ¹	Business ¹
9	United States ²	Applied Math/Stats ¹	TOEFL Total	Engineering ¹
10	US ³	Engineering ¹	CS/EECS ¹	GRE Writing %

1: School major

2: Country of last school

3: Citizenship code. Values include PR (permanent resident), FN (foreign), and US.

4: Last school degree.

The findings will help the admission committee to focus on more effective rubric measures and assign merit-based financial aid. Table 4 presents the top-10 predictors for the four classification algorithms: *Linear SVM*, *Logistic Regression*, *Random Forest*, and *Decision Trees*. These algorithms are selected because the rankings of predictive features are well-defined. In particular, for linear models the importance of a feature is proportional to the magnitude of its coefficients, while for tree-based models the ranking follows the order of the attributes used to partition the data (i.e., the attribute used to split the root node has highest rank).

Our first observation is that the *SVM* and *Logistic Regression* models rely heavily on applicants' background data including their undergraduate major, level of education, and country of origin. On the other hand, *Decision Trees* and *Random Forest* models utilize quantitative attributes such as GRE quantitative/verbal/writing scores, overall GPA, student age, and undergraduate school rankings. The superior performance of the *Random Forest* model compared to other standalone algorithms suggests quantitative measures are more reliable metrics in predicting a student's potential success in the MSDS program.

Our next analysis involves distinguishing the positive and negative predictors among the highly ranked predictive features. To this end, we resort to the magnitude of positive and negative weights provided by the linear classifiers (i.e., *SVM* and *Logistic Regression*) together with our first-hand experience in overseeing our MSDS program. Our findings suggest that students with an undergraduate major in Business, Economics, International Studies, Humanities, and Communications are poor candidates for an MSDS program, while applicants with Computer Science, Electrical Engineering, (Applied) Mathematics or Statistics backgrounds are more likely to succeed in the program. High GRE scores, Overall (undergraduate) GPA, and School Ranking, are positive indicators for success. We find these discoveries of important practical values because of the interdisciplinary nature of a data science program. Because data science programs attract students from diverse backgrounds, our studies suggest that a solid mathematics, computer science, or engineering background is essential for a student to be highly successful in an MSDS program.

5. CONCLUSION

Graduate admissions is a challenging task because it is generally controlled by faculty that have other responsibilities and priorities, and have limited training in the admissions process. The admissions process for a graduate data science program has even more challenges because it is interdisciplinary, most students do not have undergraduate backgrounds in data science, and the degree program has not existed long enough so that there is significant institutional knowledge about what applicants make the best (and worst) data science students. Thus this is an area that can benefit from data mining. The results in this paper show that mining a combination of admissions application data and student performance data can help to identify those students who are likely to do well, as well as those that are likely to struggle in their studies.

The results in this study demonstrate that our models can effectively identify both top students, who could then be offered the merit-based aid that is allocated to the MSDS program, and the bottom-performing students, who then could be denied entry into the program. The results show that our best-performing algorithms can achieve an accuracy of about 90% when identifying either the top or bottom performing students. We feel that these results are sufficiently strong that it is reasonable to take action based upon them.

The data analyses conducted in this study, as well as the examination of the features that are most important for some of the classification models, both provide valuable insight into the factors that influence success in the data science program. The key conclusions are summarized below. However, in viewing these, it is important to understand that these conclusions are based on the performance of enrolled students, so those students with weak backgrounds (e.g., very low GRE scores) will have already been excluded. Our analysis shows that the GRE quantitative score and, to a lesser extent, the GRE verbal score, do impact performance in the program, but only to a modest degree—perhaps to a lesser extent than we expected. The TOEFL score has almost no impact (partially due to the fact that all admitted students have satisfactory TOEFL scores). However, the GPA from the last degree, which is usually an undergraduate degree, has a very strong impact on performance in the program. The major associated with the last degree also plays a significant role, with computer science, mathematics, and engineering degrees positively impacting success in the program, while business, communications, economics, psychology, and humanities degrees negatively impacting performance. Students who are foreign nationals, female, hold prior graduate degrees, or who have been out of school for more than a year also tend to perform well.

There are many ways in which this study can be extended. The most straightforward is to utilize more data. Given that the current MSDS program is thriving, we expect in a few years there will be substantially more labelled data, as well as more unlabelled data since many students will either be rejected or will choose not to enroll. We believe that we can leverage this unlabelled data to improve the results via the use of semi-supervised learning algorithms. During this study, we tried to leverage the existing unlabelled data, but there simply was not enough to have a significant impact given the diversity of the applicant pool. We also tried to use our domain knowledge to form data subgroups to fine-tune the model. We hypothesize that students with different backgrounds warrant different treatment, but currently we have too little data and too many feature values to leverage this information. Exploring algorithms such as SVM+[17], which facilitates learning with heterogeneous data, can further improve the efficacy of our models. Finally, we are very interested in extending this work to other STEM graduate programs, including the MS in Cybersecurity and MS in Computer Science degree programs that currently reside in our department.

6. REFERENCES

- [1] C. M. Antons and E. N. Maltz. Expanding the role of institutional research at small private universities: A case study in enrollment management using data

- mining. *New directions for institutional research*, 2006(131):69–81, 2006.
- [2] R. Asif, A. Merceron, S. A. Ali, and N. G. Haider. Analyzing undergraduate students’ performance using educational data mining. *Computers & Education*, 113:177–194, 2017.
 - [3] B. Bakhshinategh, O. R. Zaiane, S. ElAtia, and D. Ipperciel. Educational data mining applications and tasks: A survey of the last 10 years. *Education and Information Technologies*, 23(1):537–553, 2018.
 - [4] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
 - [5] L. Chang. Applying data mining to predict college admissions yield: A case study. *New Directions for Institutional Research*, 131:53–68, 2006.
 - [6] C. Cortes and V. Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.
 - [7] T. G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.
 - [8] A. Dutt, M. A. Ismail, and T. Herawan. A systematic review on educational data mining. *Ieee Access*, 5:15991–16005, 2017.
 - [9] L. V. Fausett et al. *Fundamentals of neural networks: architectures, algorithms, and applications*, volume 3. Prentice-Hall Englewood Cliffs, 1994.
 - [10] H. C. Koh, G. Tan, et al. Data mining applications in healthcare. *Journal of healthcare information management*, 19(2):65, 2011.
 - [11] S. Lehr, H. Liu, S. Kinglesmith, A. Konyha, N. Robaszewska, and J. Medinilla. Use educational data mining to predict undergraduate retention. In *2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT)*, pages 428–430. IEEE, 2016.
 - [12] S. W. Menard. *Applied logistic regression analysis*. Number 04; e-book. 1995.
 - [13] T. Mishra, D. Kumar, and S. Gupta. Mining students’ data for prediction performance. In *2014 Fourth International Conference on Advanced Computing & Communication Technologies*, pages 255–262. IEEE, 2014.
 - [14] P. Nithya, B. Umamaheswari, and A. Umadevi. A survey on educational data mining in field of education. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 5(1):69–78, 2016.
 - [15] D. L. Olson, Y. Shi, and Y. Shi. *Introduction to business data mining*, volume 10. McGraw-Hill/Irwin Englewood Cliffs, 2007.
 - [16] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.
 - [17] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22.5:544–557, 2009.
 - [18] S. K. Yadav, S. Pal, and B. Bharadwaj. Data mining applications: A comparative study for predicting student’s performance. *International Journal of Innovative Technology and Creative Engineering*, 1(arXiv: 1202.4815):13–19, 2012.

Decomposition of Response Time to Give Better Prediction of Children's Reading Comprehension

Zhila Aghajari *
Lehigh University
Bethlehem, PA
zha219@lehigh.edu

Deniz Sonmez Unal *
University of Pittsburgh
Pittsburgh, PA
des204@pitt.edu

Mesut Erhan Unal
University of Pittsburgh
Pittsburgh, PA
meu6@pitt.edu

Ligia Gómez
Arizona State University
Tempe, AZ
ligia.gomez@asu.edu

Erin Walker
University of Pittsburgh
Pittsburgh, PA
eawalker@pitt.edu

ABSTRACT

Response time has been used as an important predictor of student performance in various models. Much of this work is based on the hypothesis that if students respond to a problem step too quickly or too slowly, they are most likely to be unsuccessful in that step. However, something that is less explored is that students may cycle through different states within a single response time and the time spent in those states may have separate effects on students' performance. The core hypothesis of this work is that identifying the different states and estimating how much time is devoted to them in a single response time period will help us predict student performance more accurately. In this work, we decompose response time into meaningful subcategories that can be indicative of helpful or harmful cognitive states. We then show how a model that is using these subcategories as predictors instead of response time as a whole outperforms both a linear and a non-linear baseline model.

Keywords

Response time, student modeling, regression models, on-task and off-task behaviors

1. INTRODUCTION

Intelligent Tutoring Systems (ITS) help students learn a wide variety of skills from problem solving [23] to reading [22, 19]. To improve ITS designs, researchers often study students' learning patterns to identify their relationship to performance and target them for intervention. Within this context, response time has been widely used to predict student performance [39, 40] and to interpret cognitive and motivational states during ITS use [39, 3, 7].

Much of the research involving response time is based on

*These authors contributed equally to this work.

Zhila Aghajari, Deniz Sonmez Unal, Mesut Erhan Unal, Ligia Gómez and Erin Walker "Decomposition of Response Time to Give Better Predictions of Children's Reading Comprehension" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 334 - 341

the hypothesis that the relationship between response time and student performance is non-linear [9]. Fast or slow response times may be indicative of both helpful and harmful cognitive states. For example, a fast response time could be a result of either mastery of a skill or guessing. Likewise, a long response time could be because of struggling or being off-task. Contextual information surrounding response time is often used in identifying the correct cognitive or motivational states. For example, a long response time after reading a bug or a hint message can be linked to reflection [32], whereas a short response time after such actions can be a sign of gaming the system [4]. Thus, previous literature has focused on identifying students' cognitive states based on sequences of actions and the time spent between them [4, 2, 7, 5]. However, students may go through different cognitive states even within a time period between consecutive actions [32]. Despite a large body of research dedicated to studying students' cognitive states, little is known about the different states a student might be in during a single response time and how time spent in those states would affect learning.

We hypothesize that response time can be divided into subcategories that can be indicative of some helpful and harmful cognitive states, and that identifying time spent on these states within one response time can improve student performance prediction. In our previous work [35], we divided response times during a reading comprehension task into two: reading and thinking time. Results of a piecewise regression model revealed that thinking time could include four states: gaming, productive thinking, wheel spinning, and mind wandering. With the insight from these results, we further investigate the different states that could occur in one response time. We compare a model that is based on decomposition of response time to a linear baseline model which only uses average response time, and also to a non-linear baseline (a piecewise regression). By decomposing the response time, we show that students can go through multiple cognitive states in between log events. We also show that by identifying how much time is devoted to these states, we can improve the predictive models of student performance.

2. RELATED WORK

2.1 Cognitive and Motivational States

Within this section we review the cognitive states that are related to productive thinking, gaming the system, and unproductive thinking. We extract these states from the broader research literature, although it should be noted that these states were also identified by teachers as important [13].

We gather learning events that are associated with robust learning under **productive thinking** behaviors. [17] divided these events into three categories: understanding and sense-making processes, induction and refinement processes, memory and fluency-related processes. Some example behaviors that fall under understanding and sense-making processes and induction and refinement processes that could be relevant in a reading domain are self-explanation and self-reflection. These behaviors are shown to be positively related to learning [32, 8].

Gaming the system is an undesirable cognitive state wherein students try to reach the correct answers and advance in the lesson by systematically misusing the features of the system [3]. It is linked to short response times and rapid actions [3]. [29] divides gaming into two main types: systematic guessing and help abuse. Systematic guessing could be inferred from short response times between step attempts [12, 28, 2], entering the same answer in multiple contexts, and entering similar answers [29]. Help abuse was defined as searching for bottom-out hints, asking for help without any reflection on the help, and entering multiple incorrect answers despite receiving help.

Within **unproductive thinking** states, we review wheel spinning and mind wandering. **Wheel spinning** occurs when the student makes an effort but does not succeed. It is linked to long response times and many help requests. [7] illustrates that if students need help solving the first twenty problems they are in wheel spinning phase, and presenting them with more problems will not be helpful. [5] showed wheel spinning is negatively correlated with flow, positively correlated with gaming and confusion, and not correlated with boredom. **Mind wandering** occurs when students involuntarily shift their attention to task-unrelated thoughts [15, 14, 34], and is associated with distraction or boredom. This cognitive state occurs 20-40% of the time during reading [30] and causes students to fail in gaining reading comprehension skills [33, 36]. As mind wandering occurs involuntarily, it is very difficult to measure, and it is often measured using self-reported approaches [24].

2.2 Response Time in Student Modeling

Response time has been widely used in different kinds of student models, and can improve the accuracy of those models [39, 20]. For example, [4] presents a model that uses response time to detect shallow learning, and [11] predicts student performance in transfer learning using response time. [6] developed an item response theory (IRT) model to show an overall level of students engagement by analyzing response times, problem difficulty, and correctness of responses. [16] also presents an IRT-based model to estimate student proficiency and motivation level where motivation was measured based on time spent between actions and a short response time was an indicator of unmotivated behavior.

In this paper, we are inspired by work that centers response

time as a non-linear predictor of students' performance. [9] suggests that the relationship between time and student success is not linear, and there is an ideal range of time for students to respond to a problem. In [10], they further support this non-linear relationship by showing that including time as a quadratic predictor instead of linear yields to a better prediction of students' performance. These studies support the intuition that accounting for the activities in different ranges of response time can give a better prediction of student performance.

Other efforts have shown success in estimating time spent on the activities that occur within a single response time. These efforts involved decomposing response time. [32] presented a model that predicts student performance relying on estimation of activities that cannot be directly observed from the log data such as thinking about hints, entering an answer, and reflecting on the hints. The preliminary results of our previous work that decomposes response times in a reading comprehension domain also revealed that students may go through multiple cognitive states during a single response time period [35].

In this work, we aim to show that identifying time spent on different cognitive states within the response time will provide better predictions of student performance.

3. CORPUS AND MEASURES

The datasets used in our work are log data collected during two studies with an iPad application called EMBRACE [38]. EMBRACE is designed to help young dual-language learners improve their reading comprehension in English. The students read interactive story books divided into chapters and they answer 3 to 9 multiple choice questions about the text at the end of each chapter. Books consisted either of narrative stories or of informational texts. Students see the text they should read in a box and they press a button labeled "Next" at the bottom of the screen to move from one sentence to another. They also see images representing what is depicted by the text.

In the full versions of EMBRACE, students are asked to either imagine the highlighted sentences or move the images on the screen to enact these sentences. They can get feedback based on how they are moving the images. Some features that are in the full versions of EMBRACE are not provided in the control version. Students still see the images in this version as well as the highlighted sentences, however, the only actions that they can perform are tapping on words to hear their pronunciations, and pressing the "Next" button to move to the next sentence. In this work, we are particularly interested in the control version as it gives us a more restricted set of student actions, which better enables us to focus on the role response time plays in reading comprehension. In the control version, we use the following measures:

1. **Student performance:** The proportion of correctly answered questions at the end of the chapters.
2. **Response time:** The time spent between when the sentence is first loaded and when the student presses the 'Next' button to proceed to the next sentence.
3. **Help requests:** The frequency of tapping on an underlined word to hear its pronunciation in a sentence.

Response time initially includes time spent on gaming, reading and thinking.

Step 1. Calculating gaming time and subtracting it from response time.

Step 2. Calculating reading time and subtracting it from remaining portion of response time.

Step 3. Distinguishing between productive and unproductive thinking times.

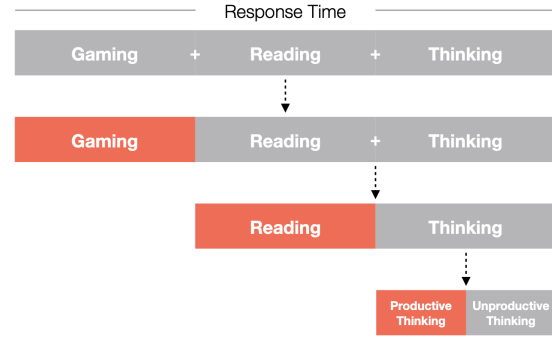


Figure 1: The protocol of designing productive vs. unproductive thinking portions in students learning.

4. **Frequency of gaming:** The “Next” button is disabled for 1 to 3 seconds depending on the length of the sentence to encourage the students to read the sentence completely. However, students might try to skip the current sentence and press this button while it is disabled. Frequency of this behavior within a sentence is our indicator of gaming since systematic and rapid actions to advance in the curriculum has been identified as gaming in previous research [3]. Note that this metric is not available in the second dataset.
5. **Decoding ability:** Decoding is defined as the ability to correctly pronounce written words. Our decoding measure is the student’s score on the decoding part of Qualitative Reading Inventory (QRI) [18] that is in range [0, 40].
6. **Sentence difficulty:** We used the Flesch-Kincaid readability grade level (FK) [21] to measure sentence difficulty. It is based on number of words in the sentence, and syllables in words. This measure represents the grade level required to understand a certain text. The difficulty of each chapter is calculated based on the average difficulty of the sentences in the chapter. FK is often used for long texts rather than single sentences. To confirm this measure is also appropriate for computing sentence level difficulty, we also computed chapter difficulty by applying FK on complete chapter texts. We did not observe a noticeable difference between computing sentence level difficulties per chapter ($M = 4.81$, $SD = 1.39$) and applying FK on complete chapter texts ($M = 4.64$, $SD = 1.35$) as $RMSE = .37$.

In our datasets, data points are distinct student-chapter pairs as student performance can only be calculated in chapter level. The first dataset includes 22 students who are native Spanish speakers from second to fourth grade with mean QRI score 34.71 ($SD = 5.19$). One student is excluded from the dataset due to having scored less than 50% on the QRI test, and thus being unable to effectively use the application. We also excluded the first chapters of the books that were read out loud to the student by the application. Finally, some of the student-chapter pairs are excluded from the dataset due to logging errors such as unrealistic response times or not completing the chapter. In total, we had data from 21 students, and 716 distinct student-chapter pairs. The mean number of book chapters per student is

Table 1: Descriptive statistics of time (in seconds) subcategories across student-chapter pairs in the first dataset (Spanish)

Measurements	Mean	SD	Min	Max
Reading Time	6.04	1.67	1.11	11.44
Productive Thinking	2.18	1.72	0	7.84
Unproductive Thinking	1.68	4.01	0	39.47
Gaming Time	0.21	0.44	0	5.36
Time Spent on Help	0.15	0.09	0.07	0.33
Time Spent on Sentence	10.12	6.02	2.75	51.5

Table 2: Descriptive statistics of time (in seconds) subcategories across student-chapter pairs in the second dataset (Mandarin)

Measurements	Mean	SD	Min	Max
Reading Time	4.40	0.65	2.67	6.12
Productive Thinking	2.68	1.07	0.01	4.09
Unproductive Thinking	1.45	3.08	0	27.79
Gaming Time	0.08	0.38	0	4.09
Time Spent on Help	0.84	0.60	0.06	3.61
Time Spent on Sentence	9.17	4.29	2.81	39.56

34.09 ($SD = 2.3$) with mean sentence difficulty 4.82 ($SD = 2.84$) across 7 story books.

In the second dataset, collected from an earlier experiment, we had 24 native Mandarin speaker students from seventh to ninth grade with mean QRI score 37.42 ($SD = 1.79$). Only one student-chapter pair was excluded from the dataset as the student in that pair did not complete the assessment task for the chapter. In this dataset we had 479 distinct student-chapter pairs. The mean number of book chapters per student is 19.95 ($SD = 0.20$) with mean sentence difficulty 4.14 ($SD = 1.06$) across 4 story books.

4. RESPONSE TIME DECOMPOSITION

Figure 1 visualizes how we decompose response time at a high level. In the following subsections we describe how each time subcategory was computed in detail. The descriptive statistics of the time subcategories for the datasets are given in Tables 1 and 2.

4.1 Time Spent on Gaming

For each sentence, if the student never pressed “Next” when it was disabled, the gaming time on that sentence is 0. Otherwise, we first calculate how long the student waited after the last time they pressed “Next” when it was disabled until they actually passed the sentence. We calculated gaming time by subtracting this waiting time from total time spent on sentence. A student who waited for a long time to pass the sentence after pressing “Next” when it was disabled will have a low gaming time estimate. Note that, in the second dataset, since our gaming indicator was not available, we did not include gaming time in our analyses.

4.2 Time Spent on Reading

We first estimated how many words students should read per minute based on their grade according to [25]. For example, if a student is in third grade, they should be able to read between 120 to 170 words per minute. To give a more specific estimate for reading rate, instead of using only the student’s grade, we include their ability in decoding English words and sentence difficulty, as students with a higher decoding ability may read more words. Similarly, in more difficult texts, students may read fewer words per minute. We first divide the normalized decoding score by the normalized sentence difficulty for each student and sentence pair. Let $[a, b]$ be the interval representing the possible values of this measure. We create another interval $[c, d]$, by getting the possible values of how many words students should read per minute within our students’ grade levels from [25]. We simply map interval $[a, b]$ on interval $[c, d]$ using the linear mapping formula below:

$$f(x) = c + ((d - c)/(b - a)) * (x - a) \quad (1)$$

Here, x is one specific decoding/difficulty score for a student-sentence pair and $f(x)$ will give an estimate for how many words this student should read adjusted by the student’s decoding ability and the difficulty of the sentence. Then, we simply calculated the time spent on reading for each sentence based on the student’s reading rate and the word count in the sentence. For example, if a student is estimated to read 120 words per minute, their reading time estimate for a 6-word sentence is 3 seconds.

$$T_{\text{read}_{u,s}} = \frac{s_w}{u_{\text{wpm}_s}} * 60s \quad (2)$$

Here $T_{\text{read}_{u,s}}$ denotes the time estimate for student u to read sentence s , s_w denotes the number of words in sentence s and u_{wpm_s} denotes the rough estimate of the reading rate for student u while reading sentence s .

4.3 Time Spent on Help Requests

We computed the exact time it takes to play the help audios. Then we computed the time spent on help requests by multiplying the time it takes to play the tapped words by two as each word is played twice.

4.4 Time Spent on Thinking

Finally, we calculate the thinking time by simply subtracting gaming time, reading time and time spent on help requests from total time spent on one sentence.

$$T_{\text{think}_{u,s}} = T_{\text{total}_{u,s}} - (T_{\text{game}_{u,s}} + T_{\text{read}_{u,s}} + T_{\text{help}_{u,s}}) \quad (3)$$

Following this procedure, thinking time was estimated to be negative for 34% of the data points as the reading time

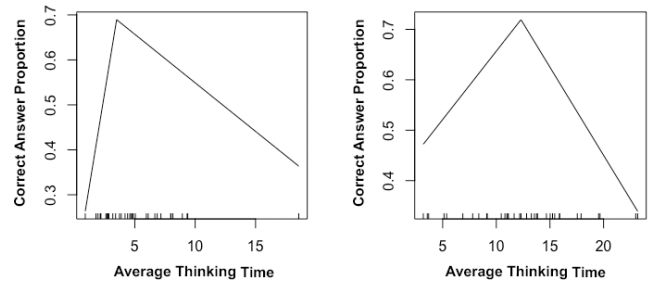


Figure 2: Example student-level thresholds for a high decoding (left) and a low decoding student (right).

estimate was higher than the total time spent on sentence. In that case, we simply adjust reading time estimate so that the time spent on sentence would be equal to reading time, and thinking time would be assigned to 0, which means that the time spent on sentence was devoted to reading and/or gaming. Even though zeroing-out negative thinking times seems to remove the variance that could be indicative of student performance, we did not observe any difference in terms of model performance. Moreover, doing so resulted in thinking time estimates becoming more interpretable.

4.5 Distinguishing Between Productive and Unproductive Thinking Time

To distinguish between productive and unproductive thinking, we use a data-driven method to find a threshold in thinking time for a student and chapter where spending more time on thinking after passing that threshold will be unhelpful. We first estimate that threshold at the student level and then similarly at the chapter level. We then combine the two thresholds to estimate one threshold for each student-chapter pair.

To find student level thresholds, using the `segmented` function in R [26, 27], we fit a separate piecewise regression model with our performance measure as the outcome and the mean time spent on thinking as the predictor for each student ($R^2 = 0.24$). There will be one breakpoint in thinking time for each student which will be independent of the chapter. Similarly, to estimate the thresholds at the chapter level, we fit one piecewise regression model with the performance measure as the outcome and the mean time spent on thinking as the predictor for each chapter (across all students) ($R^2 = 0.23$). The breakpoints represent the thresholds distinguishing between productive and unproductive thinking times for chapters. Figure 2 shows example thresholds returned from the piecewise regression models for a high decoding and a low decoding student, and Figure 3 shows example thresholds for an easy and a difficult chapter. High and low decoding students and easy and difficult chapters were decided based on median splits.

Although the separate thresholds we found are reasonable estimates, we do not use them directly when deriving the time spent on productive thinking, for two reasons. First, the threshold between productive and unproductive thinking time should be adjusted to both student and chapter characteristics in the same way that we adjusted time spent

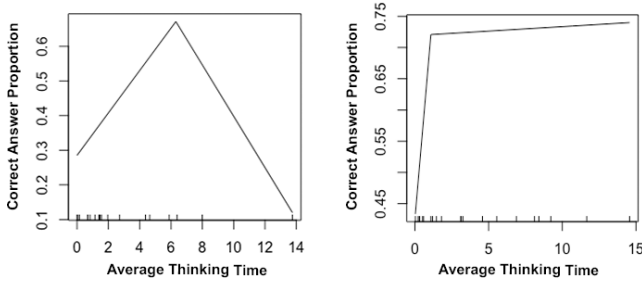


Figure 3: Example chapter-level thresholds for an easy chapter (left) and a difficult chapter (right).

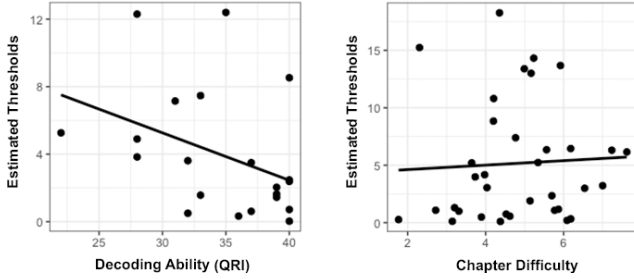


Figure 4: The relationship between decoding score and estimated student-level thresholds (left), and chapter difficulty and estimated chapter-level thresholds (right).

on reading. Second, we estimated these points by building a model which predicts the student performance, the variable that we would like to predict. Time spent on productive and unproductive thinking will be used as predictors in the model that we propose in this work. While extracting these features, leaking information from our outcome measure may cause overfitting in the final model. Therefore, we combine the thresholds found from separate regression equations. We build two separate linear regression models to predict the ‘true’ student and chapter level thresholds. Then we combine the two equations by taking their weighted average. This allows us to have one threshold estimate for an arbitrary student-chapter pair based on the decoding ability of the student and the difficulty of the chapter.

Figure 4 shows the relationship between QRI and ‘true’ student-level thresholds, and the relationship between chapter difficulty and ‘true’ chapter level thresholds. As seen in the figure, the estimated student-level thresholds are negatively correlated with decoding ability. This indicates that segregates productive and unproductive thinking regions occurs earlier for the students who scored better in the decoding test. The same figure also shows that chapter-level thresholds and chapter difficulties are far less correlated, which suggests that in estimating a threshold based on both student and chapter characteristics, the student characteristic (decoding score) is more important than the chapter characteristic (difficulty).

To combine these thresholds, we first find the equation for chapter difficulty and thinking time thresholds.

$$\lambda_{\text{chapter}} = \hat{B}_0 + \hat{B}_1 * \text{DIFF}_{\text{chapter}} \quad (4)$$

Here $\text{DIFF}_{\text{chapter}}$ denotes the chapter difficulty, and \hat{B}_1 and \hat{B}_0 are the estimated slope and y -intercept of the linear equation respectively. Similarly, we learn an equation for thinking time threshold in student level as follows:

$$\lambda_{\text{student}} = \hat{C}_0 + \hat{C}_1 * \text{QRI}_{\text{student}} \quad (5)$$

where $\text{QRI}_{\text{student}}$ denotes the student’s decoding score, and \hat{C}_1 and \hat{C}_0 denote the estimated slope and y -intercept of this linear equation respectively. We combine these two separate thresholds by taking the weighted average of them. We weigh the equations by the correlation coefficient between the QRI score and the estimated student level thresholds (x), and the correlation coefficient between chapter difficulty and the estimated chapter level thresholds (y). We find the combined threshold as follows:

$$\lambda_{\text{combined}} = \frac{x * \lambda_{\text{chapter}} + y * \lambda_{\text{student}}}{|x| + |y|} \quad (6)$$

where $x = 0.05$ and $y = -0.39$. Using this equation, we have one estimate for thinking time threshold for a given student and chapter based on both decoding ability of the student and chapter difficulty.

Finally, productive thinking time is defined as time spent on thinking before this threshold. If the time spent on thinking is less than this threshold for a given student and chapter pair, all thinking was productive and time spent on unproductive thinking is 0. If the time spent on thinking is larger than the threshold, time spent on thinking until the threshold will be counted as productive thinking time and any time beyond the threshold will be counted as unproductive thinking time.

5. PREDICTING COMPREHENSION

The core hypothesis in our work is that dividing response time into subcategories in a way that could be indicative of some helpful and harmful cognitive states will improve predictive models of student performance. To test this hypothesis, we compared the proposed linear model (Decomposed RT) to two baselines: one that uses response time as whole (Baseline 1), and another that uses response time as a non-linear predictor (Baseline 2) to show that we are not simply accounting for non-linearity in response time but we show identifying the states within response time will help us predict comprehension more accurately. We report AIC [1] and BIC [31] to show the improvement in the model is not because of the increased number of predictors. Table 3 summarizes the feature sets we used in the 3 models we compare. We performed a cross-validation at the student level within a scheme for 50 iterations in which each time we left out a unique student pair from the whole procedure (decomposition of response time and training the models) and used their data for testing.

Table 4 shows the average RMSE, R^2 , AIC and BIC values of the 50 iterations. For both datasets, we randomly flip the sign of the difference between paired model outcomes to conduct a paired-sample permutation test [37] to compare the mean of differences in evaluation metrics between our model and each baseline. We performed 1000 permutation trials in total. For the first dataset, we found significant improvements against both baselines in RMSE ($p < 0.005$), in AIC ($p < 0.001$), and in BIC ($p < 0.001$).

Table 3: Feature sets used in the models: Decomposed RT, Baseline 1, and Baseline 2. † indicates being a significant predictor ($p < 0.05$) of student performance in more than 80% of the folds.

Decomposed RT (Linear regression)	Baseline 1 (Linear regression)	Baseline 2 (Piecewise regression)
1. Frequency of gaming ¹ 2. Frequency of help requests 3. Chapter difficulty † 4. Student decoding score † 5. Time spent on reading † 6. Time spent on gaming ¹ 7. Time spent on productive thinking † 8. Time spent on unproductive thinking	1. Frequency of gaming ¹ † 2. Frequency of help requests † 3. Chapter difficulty † 4. Student decoding score † 5. Time spent on sentence †	1. Frequency of gaming ¹ † 2. Frequency of help requests 3. Chapter difficulty † 4. Student decoding score † 5. Time spent on sentence † (<i>as the non-linear parameter</i>)

Table 4: Comparison of evaluation metrics for proposed model and baselines on the first (Spanish) and second (Mandarin) dataset

	Model	RMSE	R^2	AIC	BIC
First Dataset	Decomposed RT (Linear regression)	.267 (.027)	.220 (.013)	80.242 (14.441)	124.988 (14.446)
	Baseline 1 (Linear regression)	.275 (.034)	.160 (.013)	122.399 (16.785)	153.721 (16.796)
	Baseline 2 (Piecewise regression)	.273 (.031)	.193 (.012)	100.868 (16.259)	141.139 (16.276)
Second Dataset	Decomposed RT (Linear regression)	.268 (.034)	.131 (.014)	58.350 (10.521)	91.027 (10.521)
	Baseline 1 (Linear regression)	.274 (.034)	.093 (.015)	73.090 (10.301)	97.599 (10.301)
	Baseline 2 (Piecewise regression)	.271 (.033)	.133 (.012)	57.508 (10.067)	90.185 (10.068)

For the second dataset, while decomposing response time, we made two adjustments. Firstly, since the students in this dataset were older (from 7th to 9th grade), their reading rates were adjusted for their grade level when calculating reading time. Secondly, the version of EMBRACE that was used to collect this data was not tracking when the students were pressing the “Next” button when it was disabled. Therefore, we discarded gaming time from our model. The remaining subcategories are calculated the same way as we did in the first dataset. The improvement in RMSE was significant against both Baseline 1 ($p < 0.001$) and Baseline 2 ($p < 0.005$). The improvement in AIC and BIC was significant against Baseline 1 ($p < 0.001$) while Baseline 2 was significantly better than Decomposed RT ($p < 0.05$).

Overall, Decomposed RT outperformed both baselines both in prediction error and the model fit criteria in the first dataset. However in the second dataset, although we see an improvement in prediction errors in favor of Decomposed RT, Baseline 2 had significantly better AIC and BIC values than Decomposed RT.

6. CONCLUSION

Within this paper, we proposed a new methodology to decompose response time so that time spent on gaming the system, productive thinking, and unproductive thinking states within a single response time can be accounted. Results showed that, using the time spent on these states as separate predictors rather than using response time as a whole gave better predictions of student performance. Comparison against another baseline that employs response time as a non-linear predictor also revealed that the improvement was not due to addressing the non-linearity in response time,

¹This measure is available only in the first dataset (Spanish).

and using the decomposition of response time to explain how much time was spent on different cognitive states indeed yielded better predictions. Moreover, comparison of AIC and BIC values supported that the improvement in the predictions were not due to introducing more predictors. However, we could not observe the same results for AIC and BIC between the proposed model and the non-linear baseline on the Mandarin dataset. A possible explanation is that we were not able to estimate the time spent on gaming in this dataset, thus time estimates for the other states were not as accurate as in the first dataset.

There are several other limitations of the work that need to be noted. Firstly, our estimation of reading time might not be the most accurate as there may be more factors influencing reading time than we addressed such as frequency of words and familiarity with the topic. Secondly, our model does not distinguish between the unproductive thinking behaviors (mind wandering and wheel-spinning) in its current stage. We plan to further explore how we can capture these different kinds of unproductive thinking.

In conclusion, we proposed a new method to use response time as a predictor in student modeling. The results show a promising improvement in predictive models of student performance when response time is decomposed into subcategories that can be indicative of the possible cognitive states students engage in. Future work should further assess this method’s generalizability to different student profiles and different domains.

7. ACKNOWLEDGMENTS

We thank Arthur Glenberg and Maria Adelaida Restrepo for their helpful suggestions. This work was supported by the National Science Foundation Award No. 1324807.

8. REFERENCES

- [1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected papers of hirotugu akaike*, pages 199–213. Springer, 1998.
- [2] R. S. Baker, A. T. Corbett, and K. R. Koedinger. Detecting student misuse of intelligent tutoring systems. In *International conference on intelligent tutoring systems*, pages 531–540. Springer, 2004.
- [3] R. S. Baker, A. T. Corbett, K. R. Koedinger, and A. Z. Wagner. Off-task behavior in the cognitive tutor classroom: when students’ game the system”. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 383–390, 2004.
- [4] R. S. Baker, S. M. Gowda, A. T. Corbett, and J. Ocumpaugh. Towards automatically detecting whether student learning is shallow. In *International Conference on Intelligent Tutoring Systems*, pages 444–453. Springer, 2012.
- [5] J. Beck and M. M. T. Rodrigo. Understanding wheel spinning in the context of affective factors. In *International conference on intelligent tutoring systems*, pages 162–167. Springer, 2014.
- [6] J. E. Beck. Using response times to model student disengagement. In *Proceedings of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments*, volume 20, 2004.
- [7] J. E. Beck and Y. Gong. Wheel-spinning: Students who fail to master a skill. In *International conference on artificial intelligence in education*, pages 431–440. Springer, 2013.
- [8] M. T. Chi, M. Bassok, M. W. Lewis, P. Reimann, and R. Glaser. Self-explanations: How students study and use examples in learning to solve problems. *Cognitive science*, 13(2):145–182, 1989.
- [9] I.-A. Chounta and P. Carvalho. Will time tell? exploring the relationship between step duration and student performance. International Society of the Learning Sciences, Inc.[ISLS]., 2018.
- [10] I.-A. Chounta and P. F. Carvalho. Square it up! how to model step duration when predicting student performance. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 330–334, 2019.
- [11] R. S. d Baker, S. M. Gowda, and A. T. Corbett. Towards predicting future transfer of learning. In *International Conference on Artificial Intelligence in Education*, pages 23–30. Springer, 2011.
- [12] Y. Gong, J. E. Beck, N. T. Heffernan, and E. Forbes-Summers. The fine-grained impact of gaming (?) on learning. In *International Conference on Intelligent Tutoring Systems*, pages 194–203. Springer, 2010.
- [13] K. Holstein, G. Hong, M. Tegene, B. M. McLaren, and V. Aleven. The classroom as a dashboard: co-designing wearable cognitive augmentation for k-12 teachers. In *Proceedings of the 8th international conference on learning Analytics and knowledge*, pages 79–88, 2018.
- [14] S. Hutt, J. Hardey, R. Bixler, A. Stewart, E. Risko, and S. K. D’Mello. Gaze-based detection of mind wandering during lecture viewing. *International Educational Data Mining Society*, 2017.
- [15] S. Hutt, C. Mills, S. White, P. J. Donnelly, and S. K. D’Mello. The eyes have it: Gaze-based detection of mind wandering during learning with an intelligent tutoring system. *International Educational Data Mining Society*, 2016.
- [16] J. Johns and B. Woolf. A dynamic mixture model to detect student motivation and proficiency. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 163. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
- [17] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science*, 36(5):757–798, 2012.
- [18] L. Leslie and J. S. Caldwell. *Qualitative reading inventory*. Pearson, 2016.
- [19] H. Li and W. Baer. Scaffolding adult learners’ reading strategies in the intelligent tutoring system. In *Deep Comprehension*, pages 166–179. Routledge, 2018.
- [20] C. Lin, S. Shen, and M. Chi. Incorporating student response time and tutor instructional interventions into student modeling. In *Proceedings of the 2016 Conference on user modeling adaptation and personalization*, pages 157–161, 2016.
- [21] M. Lovric. *International Encyclopedia of Statistical Science*. Springer, 2011.
- [22] K. S. McCarthy, C. Soto, C. Malbrán, L. Fonseca, M. Simian, and D. S. McNamara. istart-e: Reading comprehension strategy training for spanish speakers. In *International Conference on Artificial Intelligence in Education*, pages 215–219. Springer, 2018.
- [23] E. Melis and J. Siekmann. Activemath: An intelligent tutoring system for mathematics. In *International Conference on Artificial Intelligence and Soft Computing*, pages 91–101. Springer, 2004.
- [24] C. Mills, S. D’Mello, N. Bosch, and A. M. Olney. Mind wandering during learning with an intelligent tutoring system. In *International Conference on Artificial Intelligence in Education*, pages 267–276. Springer, 2015.
- [25] D. Morris. *Diagnosis and correction of reading problems*. Guilford Publications, 2013.
- [26] V. M. Muggeo. Estimating regression models with unknown break-points. *Statistics in medicine*, 22(19):3055–3071, 2003.
- [27] V. M. Muggeo et al. Segmented: an r package to fit regression models with broken-line relationships. *R news*, 8(1):20–25, 2008.
- [28] K. Muldner, W. Burleson, B. Van de Sande, and K. VanLehn. An analysis of students’ gaming behaviors in an intelligent tutoring system: Predictors and impacts. *User modeling and user-adapted interaction*, 21(1-2):99–135, 2011.
- [29] L. Paquette, A. M. de Carvalho, and R. S. Baker. Towards understanding expert coding of student disengagement in online learning. In *CogSci*, 2014.
- [30] J. W. Schooler. Zoning out while reading: Evidence for dissociations between experience and metaconsciousness jonathan w. schooler, erik d. reichle, and david v. halpern. *Thinking and seeing: Visual metacognition in adults and children*, 203, 2004.

- [31] G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.
- [32] B. Shih, K. R. Koedinger, and R. Scheines. A response time model for bottom-out hints as worked examples. *Handbook of educational data mining*, pages 201–212, 2011.
- [33] J. Smallwood, D. J. Fishman, and J. W. Schooler. Counting the cost of an absent mind: Mind wandering as an underrecognized influence on educational performance. *Psychonomic bulletin & review*, 14(2):230–236, 2007.
- [34] J. Smallwood and J. W. Schooler. The restless mind. *Psychological bulletin*, 132(6):946, 2006.
- [35] Sonmez Unal, Deniz. Modeling student performance and disengagement using decomposition of response time data. In *EDM. International Educational Data Mining Society (IEDMS)*, 2019.
- [36] K. K. Szpunar, S. T. Moulton, and D. L. Schacter. Mind wandering and education: from the classroom to online learning. *Frontiers in psychology*, 4:495, 2013.
- [37] H. van der Voet. Comparing the predictive accuracy of models using a simple randomization test. *Chemometrics and intelligent laboratory systems*, 25(2):313–323, 1994.
- [38] E. Walker, A. Adams, M. A. Restrepo, S. Fialko, and A. M. Glenberg. When (and how) interacting with technology-enhanced storybooks helps dual language learners. *Translational Issues in Psychological Science*, 3(1):66, 2017.
- [39] Y. Wang and N. T. Heffernan. Leveraging first response time into the knowledge tracing model. *International Educational Data Mining Society*, 2012.
- [40] X. Xiong, Z. A. Pardos, et al. An analysis of response time data for improving student performance prediction. 2011.

Whose Truth is the “Ground Truth”? College Admissions Essays and Bias in Word Vector Evaluation Methods

Noah Arthurs
Stanford University
narthurs@cs.stanford.edu

AJ Alvero
Stanford University
ajalvero@stanford.edu

ABSTRACT

Word vectors are widely used as input features in natural language processing (NLP) tasks. Researchers have found that word vectors often encode the biases of society, and steps have been taken towards debiasing the vectors themselves. However, little has been said about the fairness of the methods used to evaluate the quality of vectors. Analogical and word similarity tasks are commonplace, but both rely on purportedly ground truth statements about the semantic relationships between words (e.g. “man is to woman as king is to queen”). These analogies look reasonable when only taking into account the literal meanings of words, but two issues arise: (1) people don’t always use words in a literal sense, and (2) the same word may be used differently by different groups of people. In this paper, we split a dataset of over 800,000 college admissions essays into quartiles based on reported household income (RHI) and train sets of word vectors on each quartile. We then test these sets of vectors on common intrinsic evaluation tasks. We find that vectors trained on the essays of higher income students encode more of each task’s target semantic relationships than vectors trained on the essays of lower income students. These results hold even when controlling for word frequency. We conclude that the tasks themselves are biased towards the writing of higher income students, and we challenge the notion that there exist ground truth semantic relationships that word vectors must encode in order to be useful.

1. INTRODUCTION

Text analysis has grown into an important topic, with researchers from education, industry, social sciences, humanities, and traditional STEM programs harnessing large amounts of textual data that is widely available and relatively easy to access. Text data is usually very sparse (as most words do not appear in most documents) and difficult to use as input for mathematical models. This has given rise to a variety of vectorization methods that include simple word counting, statistical methods like TF-IDF, and neural methods used to generate dense representations called word vectors. Word

vectors have been shown to produce high quality results in a variety of machine learning (ML) and natural language processing (NLP) tasks, but this potentially comes with a social cost. Research has shown that word vectors propagate the gender and racial biases found in society [19, 7, 10]. However, little has been said about the fairness of the methods that we use to evaluate vectors.

After a set of word vectors has been trained on a corpus, researchers and engineers want to evaluate the quality of the vectors. As a result, a standard set of word vector evaluation tasks [41] has been developed in order to measure how useful and generalizable a given set of vectors is. When researchers propose new methods for training word vectors, they demonstrate the performance of their methods by evaluating the resulting vectors on these tasks. Furthermore, when NLP systems are built, vectors that perform well on these tasks are most likely to be chosen.

Word vector evaluation tasks are either intrinsic (performed directly on the vectors) or extrinsic (performed by using the vectors as inputs for a downstream task). Intrinsic evaluation is popular because it is very inexpensive, but it relies on having some secondary notion of what makes vectors useful. The most popular methods assume that there are “ground truth” semantic relationships that a set of word vectors must encode in order to be useful. However, due to sociolinguistic variation, not all language communities share the same semantic relationships [4]. As a result, in order for these tasks to be fair, they need to use semantic relationships that are universal: if semantic relationship R holds in the language patterns of group G but not group H , then the usage of R in an intrinsic evaluation task will bias researchers towards sets of vectors that model group G ’s language usage better than group H ’s. We use this framework to evaluate the fairness of two popular forms of intrinsic evaluation, analogical tasks and word similarity tasks.

When working with large text corpora, especially in educational contexts, it is important to consider the role of sociolinguistic variation [27]. In particular, students have been punished and targeted for their language practices if they are perceived to be different from the “mainstream” [38, 42]. Understanding how social variation in language affects word vectors is necessary in order to tackle two critical issues. The first is the question, “whose language is being modeled?” Word vectors are meant to capture something about the semantics of each word. If theories of sociolinguistic variation

Noah Arthurs and Aj Alvero “Whose Truth is the “Ground Truth”? College Admissions Essays and Bias in Word Vector Evaluation Methods” In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 342 - 349

tell us that people from different groups use language in different ways, then we must wonder if standard word vector sets like GloVe [35] are serving everyone equally. Second, we must ask, “how does fairness change across contexts?” In NLP, word vectors that perform well on intrinsic evaluations are used across many different contexts. However, if fairness involves taking sociolinguistic variation into account, it may not be the case that vectors that are unbiased in one context are biased in another.

Educational agencies and institutions are also increasingly relying on algorithms to help with decision-making processes. College admissions offices have been pushed to use AI [3] but have legal and ethical mandates to ensure process fairness for applicants based on their demographics and/or protected statuses, like race, gender, and religion. As the number of college applications rise and the need to hire reviewers increases, applicant admissions essays are a likely candidate for some form of automation. Research on the essays encode some degree of applicant gender and social class [2], making careful adoption of AI necessary. If sociolinguistic variation is not taken into account, algorithms have a high chance of reproducing social inequalities.

We address these issues by analyzing a corpus of over 800,000 college admissions essays (CAE) submitted to a selective, multi-campus university system. In addition to the essays, we have a variety of author metadata, including each student’s reported household income (RHI). We split the dataset into quartiles by RHI and train one set of word vectors from scratch on each quartile. After training, we find that on both the analogy and similarity tasks, the vectors trained on the writing of higher income students encode more of the target semantic relationships than vectors trained on the writing of lower income students. This indicates that the tasks can be biased against the writing of lower income students.

Our contributions are:

- to challenge the paradigm of “ground truth” labels for intrinsic evaluation by starting with the premise that language distributions vary along demographic characteristics.
- to provide a method for auditing the fairness/bias of an evaluation task, complementing existing methods for auditing the fairness/bias of word vectors themselves.
- to contribute to the educational scholarship of higher education by characterizing sociolinguistic variation in college admissions essays using established AI techniques.

2. BACKGROUND

2.1 Word Vectors

In NLP, word vectors (or word embeddings) are the standard way to translate words into input features for machine learning models. Popular word vector training algorithms like word2vec [30] and GloVe [35] are based on the distributional hypothesis, the idea that a word’s meaning is encoded in its co-occurrences with other words. In particular, word2vec tries to learn features which can be used to predict

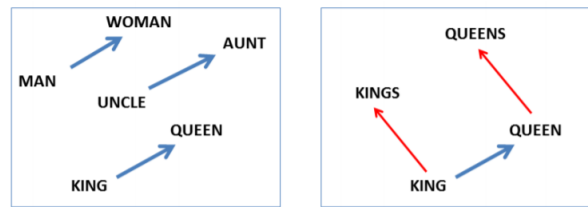


Figure 1: Illustration of simple vector operations modeling semantic (left) and syntactic (right) relationships in vector space from [32]

a word from its context (or vice versa), and GloVe trains directly from a co-occurrence matrix. Both models take in large corpora of texts and create a dense representation of every word, usually in 100- to 300-dimensional space.

2.2 Word Vector Evaluation

As described above, vectors can be evaluated intrinsically or extrinsically. Intrinsic evaluation, which is the focus of this study, involves directly examining the relationships between vectors. Intrinsic evaluation has the advantage of being much faster and more lightweight, but it comes with two downsides. The first is that intrinsic tasks do not resemble the use cases of word vectors as much as extrinsic tasks do. The second is that intrinsic tasks rely on “ground truth” human judgments about what the relationships between vectors *should* be.

The word analogy task is based on the idea that analogical relationships between words (e.g. “man is to woman as king is to queen”) should be encoded in word vectors as parallelograms (i.e. the vector that connects “man” to “woman” would be the same as the one that connects “king” to “queen”). Mathematically, this means that:

$$v_{\text{queen}} - v_{\text{woman}} \approx v_{\text{king}} - v_{\text{man}}$$

This kind of relationship has been found to hold for both for semantic (meaning-based) and syntactic (grammar-based) relationships (left and right sides of figure 1). The idea for the analogy task dates back to the 1990’s [17], but it was not proposed as a word vector evaluation technique until 2013 [31, 30]. Since then, it is common practice to compare sets of vectors on their ability to “solve” word analogies.

Word similarity is based on the idea that similar words (i.e. words that are used in similar contexts) should have similar word vectors. The word similarity task starts with a list of word pairs and involves finding the correlation between ground truth similarities between the words in each pair and the similarities between their corresponding vectors. The similarity between two vectors is in practice measured by taking their cosine similarity. The cosine similarity of two vectors \vec{a} and \vec{b} is defined as:

$$\frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|_2 \|\vec{b}\|_2}$$

which is equal to 0 if \vec{a} and \vec{b} are orthogonal, 1 if they are in the same direction, or (most of the time) something in between. The ground truth similarities, on the other hand, rely on human judgment. This task remains largely unchanged since its first iteration in 1965, when Rubenstein and Goode-nough [39] set out to test the distributional hypothesis. The big difference is that modern datasets for this task starting in 2002 with WS-353 [18] involve larger numbers of word pairs.

Both of these tasks require “ground truth” labels of some sort. The analogy task requires a list of analogies that the vectors are being tested for, while the similarity task requires ratings of the similarities between many pairs of words. These labels are problematic for two reasons. First, it has been pointed out that the labels for these tasks do not take into account the fact that words can take on many different meanings depending on context (polysemy). Second, word use and semantic intent vary along social dimensions, meaning that labels may reflect the language use of some groups better than others, thus creating bias. This second issue is the focus of our study.

2.3 Word Vector Critiques

It has been found that word vectors encode the biases present in their training data [7], and word vectors have been used to quantify the biases that exists in society [19]. Two methods have emerged for reducing bias in word embeddings: we can change our training process in order to penalize biased vectors [7], or we can identify and remove the training documents that are the source of the most bias [8].

Intrinsic evaluation methods have also fallen under scrutiny. Both the analogy task [14, 37] and the similarity task [16] have been criticized for relying on the fuzzy relationship between word similarity and vector similarity, and for not taking polysemy into account. Lastly, both tasks have been found to be poor predictors of extrinsic performance [11].

Although there have been numerous critiques of the bias encoded in word vectors and numerous critiques of intrinsic evaluation tasks, little has been said about whether or not intrinsic evaluation is biased in theory or practice. This study answers that question by identifying whether the “ground truth” semantic relationships prescribed by intrinsic evaluation tasks are shared by students of all income levels.

2.4 Sociolinguistic Variation

Language variation across spatial, demographic, and temporal dimensions is the bedrock theory behind sociolinguistics. Applied research in sociolinguistics often seeks to ameliorate systems and processes mediated through language, especially law [23] and education [36]. Relevant to this study, Bamman et al. showed significant regional variation in cosine similarity of word vectors for common words, such as “wicked” and “city” [4]. Sociolinguists are using computational methods to investigate language variation [33], but a general integration of sociolinguistics into NLP could help researchers identify and address biases.

A more equitable educational data science using text should therefore consider linguistic variation at the forefront of analysis. ML models and systems that do not account for this

risk classifying everyday language practices as hate speech, as was found to be the case with tweets written by AAVE users [40, 12]. Large datasets with student level metadata, like the data analyzed in this paper, will become increasingly common in education. Even basic sociolinguistic principles could help researchers address linguistic variation, use variation as a dependent variable, or explain how and why certain data correlates along various social dimensions to address the complicated relationship with student characteristics and language.

2.5 Household Income and College Admissions

Research on college admissions consistently shows that the college admissions process is easier for students from high income households. Studies have shown that standardized testing is strongly correlated with household income and other proxies for wealth [13], especially for black and white students. Other elements of the college application, such as financial aid forms [6] and the steps of the entire application process [26] are also more easily navigated by wealthy families than students from lower socioeconomic backgrounds. Family wealth is itself reflective of many racial and gender inequalities in the US [24].

The college admissions essay (CAE), has faced less scrutiny than standardized testing but some research has shown relationships to student identity and essay content. Using a corpus of CAE written by applicants in Britain, Jones [22] found that students from higher social classes wrote longer essays, had fewer spelling and grammatical errors, and tended to invoke markers of their higher social standing, such as the name of their elite school. Research by Kirkland & Hansen [25] found similar differences along income in diversity statement essays. They found that students from different racial backgrounds but similar socioeconomic levels wrote similar essays. Other studies have tested writing interventions with lower income students to teach them the genre of the CAE [15]. They found when students from a low income high school were explicitly trained on what they should include in their CAE, the average score of their essays on a rubric-based rating was higher than students that did not receive the intervention.

As universities move towards test optional admissions, fair analysis of CAE will become even more critical. If student backgrounds are not explicitly considered when using ML on CAE, new forms and abstractions of bias could be introduced into college admissions. However, computational methods can also shed light on potential issues of fairness in the essays. For example, Pennebaker et al. found that increased usage of function words (eg. pronouns and articles) and less personal narrative writing was positively correlated with college GPA [34].

3. DATA

The data for this study were 826,624 CAE submitted by applicants to a multi-campus, US public university system. The CAE were written across three academic years: 2015-2016, 2016-2017, and 2017-2018. These CAE were required components of the application, not additional essays submitted for honors programs, scholarships, or anything else peripheral to the main application. For this study, we removed essays that were under 100 characters and/or were

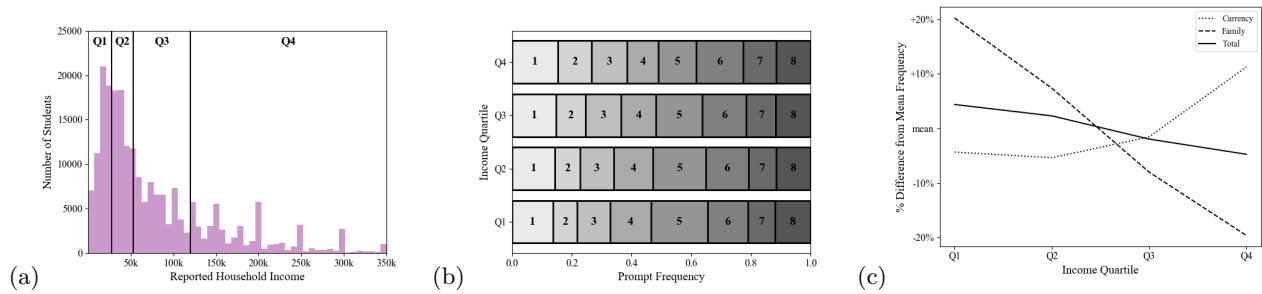


Figure 2: (a) Histogram of student RHI's with quartile boundaries marked. 8420 students with RHI >\$350k not included in plot (but included in this study). (b) Proportion of essays written for each prompt by income quartile. (c) Frequency of word usages from different subsets of the Google Analogy Test Set by income quartile. Each frequency is compared to the global mean. Total refers to all of the words in the dataset.

written by students who did not report their household income. After this filtering step, 812,020 essays remained.

3.1 Reported Household Income

A variety of metadata about each applicant and document were included as part of the dataset, but this study focuses on the reported household income (RHI) provided by each applicant. It is important to note that RHI is not an objective measure of a family's household income. When students are accepted to a university, they provide any pertinent information and documentation (such as tax return forms, W-2, etc.). However, when the application is under review before any official admission decision is made, the only income and wealth information available to a reader is the RHI.

RHI was chosen as the variable of interest for several reasons. First, language variation along class and income lines has been well established in sociolinguistic literature [5, 29]. Splitting by quartile is also a relatively crude metric, and if qualitatively and quantitatively different results emerge in the vectors across quartiles then the problem could be both fundamental and grave. For example, we might expect that the top and bottom quartile have noticeable, measurable differences, but we would not expect the second and third quartiles to be substantively different. Finally, if there are correlations between CAE and income similar to other components of the application and income, new approaches and understanding of fairness and college admissions should be considered, as well as the role of CAE in decision-making. This would push ML fairness research in college admissions to think carefully and critically about data and outcomes, as language variation is not as neat as racial or gender parity but almost always arises.

In the dataset, the average RHI is \$96,746, the median is \$53,000, and the standard deviation is \$125,000. Figure 2(a) shows distribution of income levels as well as the boundaries between the quartiles.

3.2 Prompt Choice

In 2015-2016, students had to write two personal statements to the same two prompts, meaning every applicant wrote two essays. In 2016-2017 and 2017-2018, students selected four prompts to write for from eight possible choices (70 possible combinations of prompts). The eight prompts were distinct

in theme, and if students from a certain quartile were responding to a prompt or group of prompts at significantly higher rates than students from other quartiles, our analysis could be skewed. However, figure 2(b) demonstrates that there are only mild differences in prompt choice across the income quartiles.

3.3 Word Distribution Variation

One possible source of error in this dataset is the difference in word usage between students of different backgrounds. This is a source of error because word vector training algorithms rely on large sample sizes in order to properly learn the contexts in which a given word appears. Our quartiles contain about 70 million tokens each, which is on the low end for word2vec datasets. This means that the quality of a given vector is very sensitive to that word's frequency within the data, a well-known issue that is an active topic of NLP research [20]. Practically speaking, if the word vectors trained on one quartile are able to solve an analogy that the vectors trained on another quartile fail to solve, then this could be due to the relevant words appearing more often in the first quartile.

Figure 2(c) shows the difference in word frequencies by quartile for three different subsets of the Google Analogy Test Set (GATS) [30]. We find that low income students use words from the analogy task more often overall, but this does not tell the whole story. We find that low income students use "family" words more often than high income students by a large margin, and we find (not too surprisingly) that high income students name foreign currencies more often than low income students by an even larger margin. This means that the vectors trained on the essays of low income students have an advantage on the "family" analogies, while the vectors trained on the high income students have an advantage on the "currency" analogies. We will take these word distribution-based advantages into account when analyzing the results of the analogy task.

4. METHODOLOGY

4.1 Vector Training

As mentioned above, we separately trained one set of word vectors on the writing from each income quartile. We chose to train our vectors using a word2vec Skip-Gram model in order to stay in line with Allen [1] who showed mathemati-

GATS Subset	“Viable” Analogies					“Q1 Advantage” Analogies		
	n	Q1	Q2	Q3	Q4	n	Q1	Q4
Family	420	0.629	0.733	0.702	0.681	348	0.672	0.710
Semantic	2446	0.191	0.222	0.233	0.242	391	0.609	0.645
Syntactic	9553	0.382	0.381	0.407	0.451	3307	0.433	0.488
Total	11999	0.343	0.349	0.372	0.408	3698	0.451	0.505

Table 1: Accuracy of each income quartile’s vectors on different subsets of the Google Analogy Test Set. “Viable” refers to analogies whose words appeared at least once in each training set. “Q1 Advantage” refers to viable analogies whose words appeared more often in Q1 (the essays of the lowest income students) than in Q4 (the essays of the highest income students).

cally that vectors trained in this manner would find analogies that exist in the training data.

We trained vectors of size 100 for 20 epochs using a window size of 5. We made all letters lowercase before training, but did not filter stopwords or punctuation. It is possible that changes to these hyperparameters would change the results of the study, but we feel that these are all reasonable choices given the dataset that we started with.

4.2 Vector Evaluation

For the analogy task, we use the Google Analogy Test Set (GATS) [30], which contains 19544 analogies, 8,869 of which are semantic, and 10,675 of which are syntactic. We consider a set of vectors to have “solved” the analogy “A is to B as C is to D” if the closest vector by cosine similarity to $C - A + B$ is D .

We evaluate our vectors on three similarity datasets, all of which are standard intrinsic evaluation tasks:

1. WS-353 [18] consists of 353 pairs of words along with their similarities rated on a scale from 0 to 10 by 13-16 subjects.
2. MEN [9] consists of 3000 word pairs whose similarities were determined by having subjects make binary comparisons between pairs of words rather than rating similarity directly.
3. SimLex-999 [21] consists of 999 pairs of words whose similarity was rated on a scale from 1 to 7 by 500 subjects. As opposed to the first two sets, SimLex-999 explicitly tries to avoid assigning high similarity scores to pairs of words that are associated but not similar (e.g. “coffee” and “cup”).

We measure similarity task performance using Spearman correlation.

5. RESULTS

5.1 Analogy Results

Table 1 shows the accuracy of each income quartile’s vectors on different subsets of GATS. The “Family” subset (as it is called in the original dataset) contains analogies between pairs of words that differ according to gender (e.g. “husband is to wife as grandpa is to grandma”). We chose to look at this subset in particular because it is the only semantic section of GATS whose words were used frequently by students of all four quartiles. The other semantic sections of

GATS (e.g. identifying currencies and world capitals) contained words used very infrequently by lower RHI students. We also split the entire dataset into semantic relations and syntactic relations. Semantic relations rely on word meaning (including the “Family” subset), while syntactic relations rely on morphological/grammatical differences (e.g. “bad is to worse as big is to bigger”). Finally, “Total” refers to the use of GATS in its entirety.

The first time we performed this experiment, we included all “viable” analogies (presented on the left side of Table 1). A viable analogy is one where all four words appear in each of the four sets of word vectors. With this setup, the Q1 vectors performed worst on all subsets, while the Q4 vectors performed best on all subsets except for “Family.” The difference in performance between Q1 and Q4 is very similar (5-7% of all analogies) between the semantic and syntactic subsets. This indicates that the differences we are observing are not only limited to word meaning, but to word usage as well.

Figure 2(c) shows that low RHI students use the words from GATS more frequently than high RHI students. This indicates that word distribution variation generally favors the lower RHI vectors, meaning that the higher RHI vectors performed better *despite* these variations. However, overall average word usage does not necessarily tell the whole story. It might still be the case, for example, that high RHI students use more of the words in the dataset more frequently than low RHI students. In order to more convincingly deal with the word distribution variation problem, we ran this experiment again, including what we call “Q1 advantage” analogies (presented on the right side of Table 1). An analogy has “Q1 advantage” if it is viable and its words appear more frequently in Q1 than in Q4 (i.e. the words are used more often by low RHI students).

Even when restricting ourselves to “Q1 advantage” analogies, the Q4 vectors outperform the Q1 vectors on each subset of the data, and by margins only slightly smaller than in the first experiment. This convincingly shows that word distribution variation is not to blame for the difference in performance we originally observed, as even when we *only* tested on analogies where Q1 has a word frequency advantage, the Q4 vectors solved far more of the analogies in GATS. This indicates that the observed differences in performance are due to the relationship between the analogies in GATS and the ways in which students of different quartiles use words differently.

Dataset	Q1	Q2	Q3	Q4
WS-353	0.594	0.615	0.619	0.583
MEN	0.592	0.625	0.650	0.666
SimLex-999	0.336	0.344	0.346	0.352

Table 2: Spearman correlation of each income quartile’s vectors on three word similarity tasks. Agreement with “ground truth” scores rises as income rises.

5.2 Similarity Results

Table 2 shows the results of each quartile’s vectors on each of our three word similarity tasks. Performance is reported using Spearman Correlation, although the results looked largely the same using Pearson Correlation. Note that with the exception of WS-353 (the smallest dataset), similarity task performance increased monotonically with income. This indicates that the similarity scores generated for these evaluation tasks are more in line with the way that high income students use language than the way that low income students use language. We did not filter the similarity tasks according to word frequencies, as the overall frequencies of the words in each task were very similar across the four training sets.

5.3 Qualitative Results

Word vectors by their nature pick up on semantic relationships between words [1]. It then follows that the underlying cause of these differences in intrinsic evaluation performance is a difference in word meaning between the RHI quartiles. Word vectors allow us to measure the similarity in meaning between two words in a dataset by using the cosine similarity of those two words’ vectors.

Table 3 shows the words most similar to “money” according to the Q1 and Q4 vectors. We find that while low RHI students are talking about “rent”, “expenses”, and “bills” when they talk about money, the high RHI students are talking about “savings”, “donations”, and their “allowance.”¹ This shows how a student’s experience influences the way they use language. There are probably many other words that demonstrate similar qualitative difference and variation across quartiles, but an exhaustive search through them will be considered for future study. Importantly, human readers would be able to detect the differences between the most similar vectors between Q1 and Q4, even if those differences might be subtle. For both vectors, there are clear connections to money, but the differences in how a high income student writes about money and a low income student writes about money is also clear from our qualitative assessment.

For many scholars, especially sociolinguists, the differences seen in the qualitative results alone would be firm evidence of socio-semantic variation in CAE. Research in education have consistently found that students from different social classes experience and navigate schools differently and therefore rely on different language practices to negotiate their pathways in school [28]. Sociolinguistic variation in education has therefore been widely used to study and under-

¹Though not included in the table, we found that Q2’s words were very similar to Q1’s and Q4’s words were very similar to Q3’s.

Rank	Q1		Q4	
	Word	Similarity	Word	Similarity
1	cash	0.768	funds	0.811
2	funds	0.754	fund	0.771
3	savings	0.724	monies	0.755
4	earnings	0.710	profits	0.744
5	rent	0.709	dollars	0.738
6	payment	0.705	savings	0.724
7	groceries	0.672	donations	0.701
8	expenses	0.669	donate	0.698
9	bills	0.6659	allowance	0.689
10	pay	0.659	goods	0.687

Table 3: Bills vs. Allowance: the words most (cosine) similar to “money” according to the Q1 (low income) and Q2 (high income) word vectors.

stand larger processes of social stratification and inequality. Though our qualitative analysis might not possess the depth of ethnographic research, it could still provide useful insights into how student background and experiences shape their language practices.

6. CONCLUSION

We have found that two standard intrinsic evaluation tasks (similarity and analogy) are biased against the writing of lower income students. Word vectors trained on the writing of lower income students systematically perform worse on similarity and analogy tasks than the vectors trained on the writing of higher income students. These findings do *not* indicate anything about writing quality. Rather, our results indicate that the “ground truth” semantic relationships included in these tasks are not the ground truth for everyone.

If analogies arise naturally in word vectors, then we could view the analogy task as a way of measuring what analogies exist for a given training set. If an analogy does not exist in the vectors of a given quartile, then we might say that the students who wrote those essays do not see those words as analogous. Under this perspective, our results can also serve as a way of quantifying the patterns in word usage between students of different income levels. If these patterns are not considered in large scale text analyses in education, word vectors and the many downstream tasks that use them as input could systematically bias the language practices of students based on their social class.

7. FUTURE WORK

We hope that these techniques will be used to audit other word vector evaluation tasks, both intrinsic and extrinsic. We also hope that there will be more discourse surrounding the fairness of evaluation tasks, especially given the increased use of word vectors in educational contexts. However, more work needs to be done in order to determine whether and how it is possible to debias intrinsic evaluation of word vectors with respect to various social dimensions.

Acknowledgements

We would like to thank our data providers and the Student Narrative Lab at the Stanford Graduate School of Education.

8. REFERENCES

- [1] C. Allen and T. Hospedales. Analogies explained: Towards understanding word embeddings. *arXiv preprint arXiv:1901.09813*, 2019.
- [2] A. Alvero, N. Arthurs, a. I. antonio, B. W. Domingue, B. Gebre-Medhin, S. Giebel, and M. L. Stevens. AI and holistic review: Informing human reading in college admissions. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 200–206, 2020.
- [3] E. C. Baig. Who’s going to review your college applications – a committee or a computer?, Dec 2018.
- [4] D. Bamman, C. Dyer, and N. A. Smith. Distributed representations of geographically situated language. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 828–834, 2014.
- [5] B. B. Bernstein. *Class, codes and control: Applied studies towards a sociology of language*, volume 2. Psychology Press, 2003.
- [6] E. P. Bettinger, B. T. Long, P. Oreopoulos, and L. Sanbonmatsu. The role of application assistance and information in college decisions: Results from the h&r block fafsa experiment. *The Quarterly Journal of Economics*, 127(3):1205–1242, 2012.
- [7] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.
- [8] M.-E. Brunet, C. Alkalay-Houlihan, A. Anderson, and R. Zemel. Understanding the origins of bias in word embeddings. *arXiv preprint arXiv:1810.03611*, 2018.
- [9] E. Bruni, N.-K. Tran, and M. Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47, 2014.
- [10] A. Caliskan, J. J. Bryson, and A. Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.
- [11] B. Chiu, A. Korhonen, and S. Pyysalo. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st workshop on evaluating vector-space representations for NLP*, pages 1–6, 2016.
- [12] T. Davidson, D. Bhattacharya, and I. Weber. Racial bias in hate speech and abusive language detection datasets. *arXiv preprint arXiv:1905.12516*, 2019.
- [13] E. J. Dixon-Román, H. T. Everson, and J. J. McArde. Race, poverty and sat scores: Modeling the influences of family income on black and white high school students’ sat performance. *Teachers College Record*, 115(4):1–33, 2013.
- [14] A. Drozd, A. Gladkova, and S. Matsuoka. Word embeddings, analogies, and machine learning: Beyond king-man+ woman= queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3519–3530, 2016.
- [15] J. S. Early and M. DeCosta-Smith. Making a case for college: A genre-based college admission essay intervention for underserved high school students. *Journal of Writing Research*, 2(3), 2010.
- [16] M. Faruqui, Y. Tsvetkov, P. Rastogi, and C. Dyer. Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*, 2016.
- [17] S. Federici, S. Montemagni, and V. Pirrelli. Inferring semantic similarity from distributional evidence: an analogy-based approach to word sense disambiguation. In *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, 1997.
- [18] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppín. Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1):116–131, 2002.
- [19] N. Garg, L. Schiebinger, D. Jurafsky, and J. Zou. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644, 2018.
- [20] C. Gong, D. He, X. Tan, T. Qin, L. Wang, and T.-Y. Liu. Frage: Frequency-agnostic word representation. In *Advances in neural information processing systems*, pages 1334–1345, 2018.
- [21] F. Hill, R. Reichart, and A. Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.
- [22] S. Jones. “ensure that you stand out from the crowd”: A corpus-based analysis of personal statements according to applicants’ school type. *Comparative Education Review*, 57(3):397–423, 2013.
- [23] T. Jones, J. R. Kalbfeld, R. Hancock, and R. Clark. Testifying while black: An experimental study of court reporter accuracy in transcription of african american english. *Language*, 95(2):e216–e252, 2019.
- [24] A. Killewald. Return to being black, living in the red: A race gap in wealth that goes beyond social origins. *Demography*, 50(4):1177–1195, 2013.
- [25] A. Kirkland and B. B. Hansen. “how do i bring diversity?” race and class in the college admissions essay. *Law & Society Review*, 45(1):103–138, 2011.
- [26] D. Klasik. The college application gauntlet: A systematic analysis of the steps to four-year college enrollment. *Research in Higher Education*, 53(5):506–549, 2012.
- [27] W. Labov. *Language in the inner city: Studies in the Black English vernacular*, volume 3. University of Pennsylvania Press, 1972.
- [28] A. Lareau. *Unequal childhoods: Class, race, and family life*. Univ of California Press, 2011.
- [29] D. Lawton. *Social class language and education*. Routledge, 2006.
- [30] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [31] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [32] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic

- regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013.
- [33] D. Nguyen, A. S. Doğruöz, C. P. Rosé, and F. de Jong. Computational sociolinguistics: A survey. *Computational linguistics*, 42(3):537–593, 2016.
- [34] J. W. Pennebaker, C. K. Chung, J. Frazee, G. M. Lavergne, and D. I. Beaver. When small words foretell academic success: The case of college admissions essays. *PloS one*, 9(12):e115844, 2014.
- [35] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [36] J. R. Rickford. Linguistics, education, and the ebonics firestorm. *Dialects, Englishes, creoles, and education*, pages 71–92, 2006.
- [37] A. Rogers, A. Drozd, and B. Li. The (too many) problems of analogical reasoning with word vectors. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 135–148, 2017.
- [38] J. D. Rosa. Standardization, racialization, languagelessness: Raciolinguistic ideologies across communicative contexts. *Journal of Linguistic Anthropology*, 26(2):162–183, 2016.
- [39] H. Rubenstein and J. B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.
- [40] M. Sap, D. Card, S. Gabriel, Y. Choi, and N. A. Smith. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 1668–1678, 2019.
- [41] T. Schnabel, I. Labutov, D. Mimno, and T. Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, 2015.
- [42] T. Skutnabb-Kangas and R. Dunbar. *Indigenous children’s education as linguistic genocide and a crime against humanity?: a global view*. Gáldu Kautokeino, Norway, 2010.

A Dataset of Learnersourced Explanations from an Online Peer Instruction Environment

Sameer Bhatnagar
Polytechnique Montreal
sameer.bhatnagar@polymtl.ca

Amal Zouaq
Polytechnique Montréal

Michel C. Desmarais
Polytechnique Montréal

Elizabeth Charles
Dawson College

ABSTRACT

Online *Peer Instruction* has become prevalent in many “flipped classroom” settings, yet little work has been done to examine the content students generate in such a learning environment. This study characterizes a dataset generated by an open-source, web-based homework system that prompts students to first answer questions, and then provide explanations of their reasoning. Of particular interest in this dataset, is that students are also prompted to evaluate a subset of peer explanations based on how convincing they are, as part of the Peer Instruction learning script. Since these student “votes” are then used in the selection of what is shown to future learners, we cast this as an instance of *learnersourcing*, a paradigm that presents new research opportunities for the Learning Analytics community. This study characterizes a dataset from one *Peer Instruction* tool, that includes not only the student generated answers and explanations, but this novel “vote” attribute, which aims to capture how convincing each explanation is to other learners. The dataset includes longitudinal observations of student responses over the course of a semester, following groups from three STEM disciplines. The data is made available to interested researchers¹.

Keywords

datasets, learnersourcing, peer instruction

1. INTRODUCTION

The effectiveness of *Peer Instruction* on learning [4] [25] *in-class*, and the success of Intelligent Tutoring Systems and MOOCs *outside* of class, have in part, led to the development of web-based platforms for *asynchronous* Peer Instruction [6][27]. Recently, other similar learning environments have been developed, centred on having students explain their reasoning, and then evaluate the explanations of

their peers[22][5]. The increasing use of this form of online learning exercise implies that a new type of data is being generated, wherein lie opportunities to examine theories of how self-explanation and comparative peer assessment may impact learning.

There are several pragmatic motivations for extending Peer Instruction to out-of class activities. First, when scaling-up *Just-in-Time-Teaching* environments, a web-based platform for asynchronous peer instruction can substantially reduce the time teachers’ spend searching the data to identify student misconceptions. Second, when students are asked to compare answers with peers, they receive a form of immediate feedback on their own explanation. Last, when posting threads and sub-threads to large scale on-line discussions, such as MOOCs, an asynchronous Peer Instruction platform offers a more structured alternative and ties student explanations to an answer choice, allowing for more organized aggregation of ideas [2].

These platforms open new research questions and opportunities for the Educational Data Mining community. First, these systems capture new modalities of data, specifically, the written explanations for answer choices, which acts as proxy data: representing the cognitive reflections elicited in conversations students have with peers during small- group in-class Peer Instruction discussions. Second, these environments introduce challenges common to any platform centred on user generated content: quality control and recommendation. The power of having students generating the explanations to different answer choices, and then rating them, enables scaling up of technologies that facilitate flipped teaching practices[14]. However, once these tools *do* scale, sheer volume requires automatic approaches for filtering out low-quality content. Once filtering is complete, recommendation algorithms need to be in place to most effectively help current students navigate the large volume of content generated by past students, with the ultimate objective of optimizing individual learning gains. Further research is needed on *learnersourced* data sets so as to develop best practices that leverage the effectiveness of student written and ranked explanations for adaptive learning experiences, while avoiding the pitfalls that can lead to the valuable data drowning in noise.

2. OBJECTIVES

¹account required at <https://myDALITE.org/signup>

This paper characterizes a dataset generated inside one on-line platform for asynchronous Peer Instruction, with the aim of identifying the potential research questions and limitations afforded by this novel application. The use of the tool is growing, now reaching over 50 course offerings across at least 5 undergraduate institutions in different science disciplines. The contexts are varied, but the common thread is that instructors are all using the tool as an attempt to increase in-class student engagement with pre-instructional quizzes, and tailoring their lectures based on the free-text explanations students provide for their answers. Thus, the ultimate goal of this study is two-fold:

- introduce a novel source of data to the Educational Data Mining research community, which has the potential to open new lines of inquiry based on the unique “voting” attribute. Students not only write explanations to justify their answer choice to conceptual science questions, but they are asked to choose which of a subset of their peers’ explanations are most convincing.
- identify opportunities and challenges related to the design of platforms that rely on *learnersourced* content, such as choosing the most effective content to foster learning; filtering weak or irrelevant student explanations; categorizing and summarizing student explanations for teacher reporting in large classes.

3. BACKGROUND AND RELATED WORK

3.1 Peer Instruction

The interactive engagement technique most relevant to our work here is Peer Instruction: a method for promoting classroom discussion that has been shown to enhance learning [8]. In this common classroom practice, teachers

1. poll their students on a multiple-choice item, using some form of Audience Response System (e.g. clickers),
2. collect the distribution of answers, and maybe even share back with the students,
3. without revealing the correct answer, prompt students to explain their reasoning for their answer choice to a peer nearby, ideally with someone with a different perspective
4. re-poll the students after the small group discussions.

The platform at the centre of this study facilitates an *asynchronous* version of the above script.

3.2 Comparison-based peer assessment

There are other systems similar in design to asynchronous peer instruction; they differ in that the items prompt for open-ended responses, as opposed to multiple-choice questions. However these systems still include a similar *review*-step after submission of an answer, where students are asked to compare and evaluate the quality of the explanations submitted by peers who had already answered the item.

For example, in the ComPAIR system [22], students first submit their written answer to a prompt. They are then shown pairs of their peers’ answers, prompted first to give feedback to each of the answers in the pair, and then choose

one as the better response. The pairwise comparison at the heart of this tool leverages learners’ inherent ability to make judgments regarding an answer’s quality *relative to another*, to make up for the lack of expertise usually needed to provide useful feedback on content in isolation. JuxtaPeer [5] is a similar system, where the pairwise comparisons are anchored on one object at a time, and have been shown to improve the quality of feedback that peers can provide to one another.

3.3 Explanations Datasets

Two of the most prominent sources of learning analytics datasets are from the ASSISTments platform[15], and PSLC DataShop[19]. They both provide significant contributions to Learning Analytics and Educational Data Mining researchers, by making available a wide variety of data from different on-line learning tools. They include datasets with free text responses, including math hints generated by students in ASSISTments, and student explanations to science questions inside the Andes project (hosted in DataShop).

If casting student explanations as *short arguments* in favour of their answer choice, we can look to the Argumentation Mining research community for sample datasets. For example, a dataset of persuasive student essays that are fully annotated for argumentative relations was recently released [26]. The International Corpus of Learner English[12] is used extensively to model how students make arguments.

Another sign of the growing interest in analyzing student generated text are the Automated Essay Scoring[16] and Automatic Short Answer Scoring [17] competitions hosted on the data science platform, kaggle.com. These datasets are still freely available as well.

To the best of our knowledge, none of the above data sources include all of the defining characteristics that are generated by online Peer Instruction, such as the student’s initial answer choice and explanation, a student’s second answer choice after having reviewed peer explanations, and most importantly, the peer explanation the student found most convincing.

3.4 Learnersourcing explanations

Web-based homework systems are effective because students get immediate feedback as to whether they answered correctly. However, as the number of question items grows, as well as associated answer choices, generating high quality explanations that help different types of learners resolve different sets of misconceptions, is impractical for teachers [14]. Moreover, explanations written by content experts may also suffer from the *expert blind spot*, wherein their high level of familiarity with the subject matter actually might actually make their explanations more difficult to understand to novices [20].

The concept of *learnersourcing* is a sub-type of *crowdsourcing*, wherein domain novices contribute to the human computation workflow as part of their learning process [28]. PeerWise [10] is an environment within which students make their own questions, and share them with peers, along with accompanying solutions. RiPPLE is a tool that follows the same model, but adds an adaptive recommendation engine [18]. The AXIS system [29] prompts students to provide ex-

Thin lens 1

A converging lens causes a real image to project, inverted, onto a screen. If the lower half of the lens is completely covered...

- ☒ A. The top half of the real image is missing
- ☐ B. The lower half of the real image is missing
- ☐ C. The section of the real image that is visible depends on the angle you view the image with
- ☐ D. The full real image does form, but it is dimmer than before
- ☐ E. There will be no image formed on the screen

Rationale*

The light rays from the bottom half of the object would normally end up at top half of real image, but now those are being blocked!

Figure 1: Asynchronous Peer Instruction - Step 1, screenshot of student answering multiple-choice question, and explaining their thinking inside a text box

planations to their answers, rate the explanations of their peers, and then machine learning to curate these to a constantly evolving set of explanations that optimize for promoting student learning. ASSISTments, another widely used learning platform, developed the PeerASSIST plugin [24], which asked students to write explanations to their answer submissions, to be used as hints for future students.

4. MYDALITE: PLATFORM AND DATA

4.1 The Platform

dalite-ng is an open-source project [23] that has been in active development since 2013, and has been used in MOOCs as well as on campus course offerings. myDALITE.org is one instance of this code-base, offered as a hosted service that is free to all teachers and students. It is maintained by a network of learning science researchers and practitioners, whose mission is to promote the uptake of student-centred active learning pedagogical practices. Teachers sign up, author their own questions, and distribute to their students at their discretion. The script for the student completing a question item in *dalite-ng* is:

1. **Question start:** student is presented with a multiple-choice question. They are asked to choose an answer choice and enter a free text response to explain their reasoning.
2. **Question review:** without indicating whether the student chose the correct answer, the tool reflects back to the student their own choice, and the explanation they just entered. They are then prompted to reconsider their answer, by reading the explanations of other students. In the top half of the page, they are shown up to 4 other explanations by students who chose the same answer choice. In the second half of the page, they are then shown up to 4 more explanations to a different answer choice. Students must indicate which is their second answer choice in this re-

Thin lens 1

A converging lens causes a real image to project, inverted, onto a screen. If the lower half of the lens is completely covered...

A. The top half of the real image is missing

B. The lower half of the real image is missing

C. The section of the real image that is visible depends on the angle you view the image with

D. The full real image does form, but it is dimmer than before

E. There will be no image formed on the screen

You answered **A** and gave this rationale:

The light rays from the bottom half of the object would normally end up at top half of real image, but now those are being blocked!

Consider the problem again, noting the rationales below that have been provided by other students. They may, or may not, cause you to reconsider your answer. Read them and select your final answer.

- A. ☐ The image is inverted therefore the top half of the original image is on the bottom half of the image formed on the screen. If we cover the bottom half of the screen we cannot see the top half of the original image.
- ☒ B. Since the image is inverted, the bottom part that is covered would have been placed at the top. And since it is covered, that part will be missing in the final image.
- ☐ C. The bottom rays will not pass through the lens, but the top rays will. Since the final real image is inverted, then only the bottom part of the image will be present (representing the top part of the object).
- ☐ D. By blocking the lower half of the lens, you block the rays that end up forming the top half of the image (note that the image is inverted!); therefore the top half will be missing
- ☐ E. I stick with my own rationale.

- D. ☐ Clearly not all rays will hit the screen, but enough rays emerging from all of the object WILL hit the screen. The final real image will be complete, but will be less bright (hence dimmer) because not all of the light intensity goes through the lens
- ☐ The image will still form however it will be dimmer than the original if was covered since there would be more light coming in if there was nothing covering it
- ☐ by covering the lens you only dim down the image you are not decreasing the actual object itself
- ☐ the light image wont be as bright since it escapes a little around the lenses

Figure 2: Asynchronous Peer Instruction - Step 2 , screenshot of student choosing a peer's explanation of a different answerchoice

view step, by selecting one of these explanations. They also have the option of selecting their own explanation as the most convincing. There are several factors that go into the selection of what the students are shown here:

- if the student answered incorrectly on the first step, the explanations in the second half of the page will be for the correct choice
 - if the student did in fact answer correctly on the first step, the explanations in the second half of the page will be for the most popular incorrect answer.
 - There are two different heuristics for the selection of explanations for each answer choice:
 - Random, which is useful for when a question is newly introduced to the database, and not enough students have answered to reliably estimate which answers are most *convincing*
 - preferentially selecting from explanations that have already been chosen as convincing
3. **Question summary** The entire flow of information is reflected back to the student for review: their first answer, their own explanation, their second answer

choice and the associated explanation that they chose as most convincing. The correct answer is also finally revealed.

4.2 Data Collection

The data in this study comes from the 2018-2019 academic year, wherein the platform was more heavily used than ever before, due to additional on-boarding support offered to teachers by the host network. All teachers who make question items on the platform must release their content under Creative Commons licenses, and are made aware that the learning-data generated by the students in their groups may be used for academic research. Students are advised upon signing in, that their learning traces, in anonymized form, may be used for research, and that if they do not wish to share their data, they can revoke their consent at any time, without any impact on academic standing in their courses. The data gathered for this study spans three STEM disciplines where there happened to be the most activity: Biology, Chemistry, and Physics. There are many different groups of students in Physics and Chemistry, each with a different teacher (although all in undergraduate level courses), while all of the data in Biology comes from one large freshman group that used the tool very heavily. In the case of a few groups, the items were assigned by teachers as optional, not-for-credit items, meant to provide extra practice study exercises (this information is provided in the meta-data file of the dataset). For those cases when myDALITE was used for credit, students received 0.5 marks for choosing the correct answer on their first attempt, and 0.5 marks for choosing the correct answer choice after the review step. No credit was ever assigned based on a formal expert evaluation of the student explanations.

4.3 Dataset

Each record in the dataset is comprised of the following fields:

- anonymized student identifier
- anonymized group/course identifier (with meta-data on whether the activities were assigned for credit or not)
- question prompt text (and any associated images)
- student's first answer choice
- student's explanation for their first answer choice
- peer explanations shown to student on second step
- student's second answer choice
- the peer explanation they selected as most convincing for their second answer choice
- timestamps associated with
 - when the student first opened the problem
 - when the student entered their first answer choice, and associated explanation
 - when the student entered their second answer choice, and associated peer explanation

Certain filters were applied for the purposes of data extraction for this study. The only groups that were retained were those having 10 students or more, each of whom having answered at least 10 questions. The only disciplines that were included in the current dataset were ones with over 10,000 student responses.

Table 1: Size of dataset across disciplines

	N_g	N	N_q	N_a	\overline{N}_a
Biology	1	346	232	19653	57
Physics	16	1250	572	50286	40
Chemistry	16	1055	532	28319	27

Table 2: Relative number of answer transitions, from step 1 to step 2

	$1^{st}C$	Δ			Δe
		\sum	$r \rightarrow w$	$w \rightarrow r$	
Biology	0.70	0.10	0.01	0.09	0.40
Physics	0.79	0.09	0.01	0.07	0.44
Chemistry	0.69	0.12	0.01	0.11	0.38

5. DESCRIPTIVE STATISTICS

As can be seen in Table 1, there are relatively similar numbers of responses across the three disciplines.

- N_g : number of groups
- N : number of unique students
- N_q : number of items
- N_a : total number of answers by all students
- \overline{N}_a : average number of items completed by each student

This table demonstrates the valuable longitudinal nature of the dataset, in that across the disciplines, there are, on average, more than 25 observations per student, which could help building a more robust learner models.

In Table 2, we see the proportion of times students changed their answers on the answer review step.

- $1^{st}C$: fraction of responses where students chose the correct answer choice on their first attempt
- Δ : fraction of responses where students switch their answer choice on review step
 - \sum : total fraction of answers where students changed their answer choice from step 1 to step 2
 - $r \rightarrow w$: fraction of responses where students switch their answer choice on review step, going from right to wrong
 - $w \rightarrow r$: fraction of responses where students switch their answer choice on review step, going from wrong to right, presumably after reading their peers' explanations
- Δe : fraction of responses when students do not change their answer choice on review step, but choose an explanation other than their own as most convincing

Across the disciplines, the items in this dataset are easy enough for students to choose the correct answer choice on their first attempt almost three out of four times. The explanations of their peers are almost never able these convince students to switch from the right answer choice to a wrong one. However of the students who choose the wrong answer on their first attempt, after having access to the explanation of their peers, these students switch to the correct answer choice at the review step almost one out of three times. These

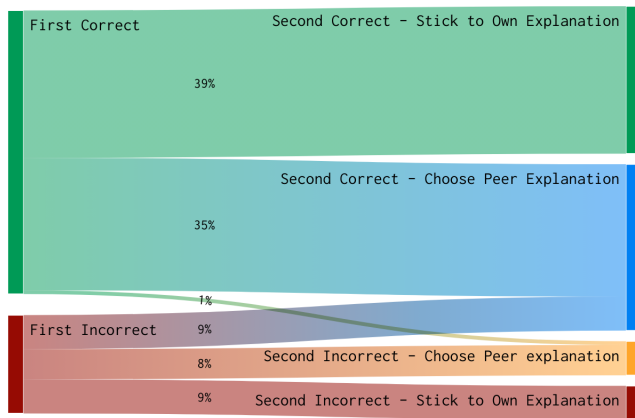


Figure 3: Visualizing student transitions in asynchronous peer instruction with a Sankey diagram

Table 3: Descriptive statistics on explanation word counts

	\overline{WC}	$WC < 5$	$\Delta e_{\text{longest}}$
Biology	7	0.63	0.19
Physics	15	0.42	0.23
Chemistry	19	0.19	0.18

relative transitions are visualized in the Sankey diagram in Figure 3.

In table 3,

- \overline{WC} is the average word count of the explanations
- $WC < 5$ is the proportion of explanations that have less than 5 words
- $\Delta e_{\text{longest}}$ the proportion of times students selected the peer explanation that had the most words amongst those that they were shown.

Here in Table 3, we see that many students write explanations that are too short to form a sentence in the Biology subset, and that even in the other disciplines, the explanations are not long-form persuasive essays, but likely closer to short answers. However, students seem to show a preference for explanations that are longer in length when “voting” for the most convincing explanation on the review step.

6. DISCUSSION

Learnersourcing shows immense potential for scaling up online Peer Instruction, but also presents new challenges common to contexts centred on user-generated content.

A quick sampling of the large number of explanations with less than 5 words likely indicates that students do not see the value of writing explanations, unless they will receive course credit for the task. Work from the argument mining community may be useful here to automatically assess the quality of explanations. Under study is the impact of web-based reputation systems on increasing student engagement, which have been shown to increase engagement in learning environments by offering virtual achievement rewards,

such as badges and leaderboards[9]. Another open research question is in automatic quality control, given that the first few students who complete a question, and submit an explanation, will have their work shown to many subsequent students. Work that has been done on automatic filtering [11] of explanations based on unsupervised clustering could prove beneficial here.

The value and uniqueness of this dataset remains in the “voting” data: modelling what linguistic properties and conceptual constructs students find convincing, in the language of their peers, is fertile ground for research. The longitudinal data also allows for modelling the evolution of how students start integrating domain specific concepts into their explanations across a semester, as well as “voting” for them in the peer-explanations they find most convincing.

6.1 Future Work

Work must now be done on better understanding how to optimize the heuristics that select what peer explanations are shown to students in order to enhance learning. This will require building student models of ability and models of item difficulty. The linguistic properties are also of key interest: can this mode of comparative peer assessment data be used to inform our models of whether students have attained domain literacy? Finally, how do such environments promote student engagement in *flipped classroom* contexts? We look forward to collaborating with the community through this novel source of data to along these lines of research.

Many of the design/implementation decisions for these platforms are made with pragmatic motivations in mind and need to be better informed by learning analytics theory. The platform at the center of this study is a model to examine more closely also because it is an open-source project, developed as part of Research Practice Partnership [7], where learning analytics researchers are actively working with instructors using the tool to better align teaching practices with sound pedagogical design.

7. ACKNOWLEDGMENTS

Funding for the development of myDALITE.org is made possible by *Entente-Canada-Quebec* program, and the *Ministère de l’Éducation et Enseignement Supérieure du Québec*. Funding for this research was made possible by the support of the Canadian Social Sciences and Humanities Research Council *Insight* Grant. Special thanks to Dr. Jonathon Sumner for help with data visualization. This project would not have been possible without the SALTISE/S4 network of researcher practitioners, and the students using myDALITE.org who consented to share their learning traces with the research community.

References

- [1] S. Bhatnagar, M. Desmarais, C. Whittaker, N. Lasry, M. Dugdale, and E. S. Charles. An analysis of peer-submitted and peer-reviewed answer rationales, in an asynchronous Peer Instruction based learning environment. In O. C. Santos, J. G. Boticario, C. Romero, M. Pechenizkiy, A. Merceron, P. Mitros, J. M. Luna, C. Mihaescu, P. Moreno, A. Hershkovitz, S. Ventura, and M. Desmarais, editors, *Proceedings of the 8th International Conference on Educational Data Mining*, 2015.

- [2] S. Bhatnagar, N. Lasry, M. Desmarais, and E. Charles. DALITE: Asynchronous Peer Instruction for MOOCs. In *European Conference on Technology Enhanced Learning*, pages 505–508. Springer, 2016.
- [3] J. L. Bishop, M. A. Verleger, and others. The flipped classroom: A survey of the research. In *ASEE national conference proceedings, Atlanta, GA*, volume 30, pages 1–18, 2013.
- [4] J. E. Caldwell. Clickers in the large classroom: Current research and best-practice tips. *CBE—Life Sciences Education*, 6(1):9–20, 2007.
- [5] J. Cambre, S. Klemmer, and C. Kulkarni. Juxtapeer: Comparative Peer Review Yields Higher Quality Feedback and Promotes Deeper Reflection. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–13, Montreal QC, Canada, 2018. ACM Press.
- [6] E. S. Charles, N. Lasry, C. Whittaker, M. Dugdale, K. Lenton, S. Bhatnagar, and J. Guillemette. Beyond and Within Classroom Walls: Designing Principled Pedagogical Tools for Student and Faculty Uptake. International Society of the Learning Sciences, Inc.[ISLS]., 2015.
- [7] C. E. Coburn and W. R. Penuel. Research–practice partnerships in education: Outcomes, dynamics, and open questions. *Educational Researcher*, 45(1):48–54, 2016.
- [8] C. H. Crouch and E. Mazur. Peer instruction: Ten years of experience and results. *American Journal of Physics*, 69(9):970–977, 2001.
- [9] P. Denny. The Effect of Virtual Achievements on Student Engagement. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 763–772, New York, NY, USA, 2013. ACM. event-place: Paris, France.
- [10] P. Denny, J. Hamer, A. Luxton-Reilly, and H. Purchase. PeerWise: Students Sharing Their Multiple Choice Questions. In *Proceedings of the Fourth International Workshop on Computing Education Research*, ICER '08, pages 51–58, New York, NY, USA, 2008. ACM. event-place: Sydney, Australia.
- [11] V. Gagnon, A. Labrie, M. Desmarais, and S. Bhatnagar. Filtering non-relevant short answers in peer learning applications. In *Proc. Conference on Educational Data Mining (EDM)*, 2019.
- [12] S. Granger, E. Dagneaux, F. Meunier, and M. Paquot. *International corpus of learner English*. 2009.
- [13] R. R. Hake. Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American journal of Physics*, 66(1):64–74, 1998.
- [14] N. Heffernan, K. Ostrow, K. Kelly, D. Selent, E. Inwegen, X. Xiong, and J. Williams. The Future of Adaptive Learning: Does the Crowd Hold the Key? *International Journal of Artificial Intelligence in Education*, 26, 2016.
- [15] N. T. Heffernan and C. L. Heffernan. The ASSISTments Ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.
- [16] Kaggle. The Hewlett Foundation: Automated Essay Scoring, 2012.
- [17] Kaggle. The Hewlett Foundation: Short Answer Scoring, 2013.
- [18] H. Khosravi. Recommendation in personalised peer-learning environments. *arXiv preprint arXiv:1712.03077*, 2017.
- [19] K. R. Koedinger, R. S. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the EDM community: The PSLC DataShop. *Handbook of educational data mining*, 43:43–56, 2010.
- [20] M. J. Nathan, K. R. Koedinger, M. W. Alibali, and others. Expert blind spot: When content knowledge eclipses pedagogical content knowledge. In *Proceedings of the third international conference on cognitive science*, volume 644648, 2001.
- [21] G. M. Novak. Just-in-time teaching. *New directions for teaching and learning*, 2011(128):63–73, 2011.
- [22] T. Potter, L. Englund, J. Charbonneau, M. T. MacLean, J. Newell, I. Roll, and others. ComPAIR: A new online tool using adaptive comparative judgement to support learning with peer feedback. *Teaching & Learning Inquiry*, 5(2):89–113, 2017.
- [23] SALTISE. SALTISES4/dalite-ng, June 2019. original-date: 2018-01-11T16:36:55Z.
- [24] D. Selent. *Creating Systems and Applying Large-Scale Methods to Improve Student Remediation in Online Tutoring Systems in Real-time and at Scale*. PhD thesis, Worcester Polytechnic Institute Polytechnique Institute, June 2017.
- [25] M. K. Smith, W. B. Wood, W. K. Adams, C. Wieman, J. K. Knight, N. Guild, and T. T. Su. Why peer discussion improves student performance on in-class concept questions. *Science*, 323(5910):122–124, 2009.
- [26] C. Stab and I. Gurevych. Annotating Argument Components and Relations in Persuasive Essays. In *COLING*, pages 1501–1510, 2014.
- [27] T. . L. T. Univeristy of British Columbia. ubc/ubcpi, Aug. 2019. original-date: 2015-02-17T21:37:02Z.
- [28] S. Weir, J. Kim, K. Z. Gajos, and R. C. Miller. Learnersourcing subgoal labels for how-to videos. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 405–416. ACM, 2015.
- [29] J. J. Williams, J. Kim, A. Rafferty, S. Maldonado, K. Z. Gajos, W. S. Lasecki, and N. Heffernan. AXIS: Generating Explanations at Scale with Learnersourcing and Machine Learning. In *Proceedings of the Third (2016) ACM Conference on Learning @ Scale - L@S '16*, pages 379–388, Edinburgh, Scotland, UK, 2016. ACM Press.

Effective Forum Curation via Multi-task Learning

Faeze Brahman
University of California
Santa Cruz
fbrahman@ucsc.edu

Nikhil Varghese
University of California
Santa Cruz
nivarghe@ucsc.edu

Suma Bhat
University of Illinois at
Urbana-Champaign
spbhat2@illinois.edu

Snigdha Chaturvedi
University of North Carolina at
Chapel Hill
snigdha@cs.unc.edu

ABSTRACT

Despite several advantages of online education, lack of effective student-instructor interaction, especially when students need timely help, poses significant pedagogical challenges. Motivated by this, we address the problems of automatically identifying posts that express confusion or urgency from Massive Open Online Course (MOOC) forums. To this end, we first investigate the extent to which the tasks of confusion detection and urgency detection are correlated so as to explore the possibility of utilizing a multitasking set-up. We then propose two LSTM-based multitask learning frameworks to leverage shared information and transfer knowledge across these related tasks. Our experiments demonstrate that the approaches improve over single-task models. Our best-performing model is especially useful in identifying posts that express both confusion and urgency, which can be of particular relevance for forum curation.

1. INTRODUCTION

Massive online courses have changed the academic landscape of today, offering convenient alternatives to learners at significantly reduced costs, compared to traditional educational institutions. With more than six million students taking at least one online course as part of their degree program [16], online education has already become one of the most popular higher education supplements.

Despite several advantages associated with online education, such as diversity of programs, lower cost, and more flexible learning environment, factors such as lack of personalization and low instructor-student ratio pose significant challenges to this learning environment. For the most part, discussion forums continue to be the sole platform for student interaction with others (students and instructors), where learners share their ideas, opinions, or even express their concerns and questions about the course material. Unfortunately, in a typical online class, these forums can quickly get difficult

to manage with few instructors and several learners getting involved and posting their concerns. This situation can hamper the instructors' ability to gauge students' comprehension of course materials and address students' concerns in a timely manner, ultimately reducing learning effectiveness for students.

One way of bringing about the much needed immediacy is by way of automatic curation of the forums, where posts related to confusion about the course material, or those that raise urgent issues are automatically identified. For instance, identifying posts that express confusion (*Confusion Detection*) could help instructors in adapting their teaching strategies during the course by employing more examples, altering the course syllabus or slowing down the pace of instruction. Likewise, automatically identifying urgent posts, i.e. posts which need an immediate response (*Urgency Detection*) and resolving them in a timely manner is important for keeping students engaged. The two types of posts are related but different in the sense that posts that express confusion seek help about the content of the course material while posts that express urgency also seek help but not necessarily directly about the course content. Nevertheless, the ultimate goal of both types of posts is to seek help from others and so there is promise in designing methods that can learn them simultaneously in a multi-tasking set-up.

While previous works have focused on addressing a single forum curation task [1, 20, 21, 22], other studies [24, 25] have also shown that learning features that help address one task may be gainfully used for other tasks—an aspect central to a multi-task learning framework. Another reason for exploring multi-task learning in this domain is the limited availability of labeled data. The use of supervised machine learning approaches requires labeled data annotated by experts, which can be time-consuming, costly, and difficult to obtain in this domain. Unlike single-task frameworks which often suffer from insufficient annotated data, the proposed multi-task framework can share information between related tasks leveraging beneficial information, thus avoiding the need to have large amount of labeled data for individual tasks. However, this comes at the cost of increased model-parameters, which can instead hurt the model. Also, if the jointly learned tasks are weakly correlated, it might be more fruitful to focus on one task at a time since multi-tasking might introduce more noise than useful signals. Despite these issues,

Faeze Brahman, Nikhil Varghese, Suma Bhat and Snigdha Chaturvedi "Effective Forum Curation via Multi-task Learning" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 356 - 363

the potential gains via an implicit increase in the sample size for training our model by making it learn related tasks has the promise of averaging the noise of each task and thus improving generalization.

In this paper, we propose two multi-task learning architectures, namely Shared-BiLSTM and Specific-Shared Multi-Task, based on Long Short Term Memory (LSTM) networks. Our goal is to use these architectures for forum curation by jointly learning the tasks of Confusion detection and Urgency detection. To investigate the potential promise of our approaches, in light of the concerns mentioned above, we design experiments to answer the following research questions:

RQ1: To what extent are different tasks in this domain correlated?

RQ2: What is an effective multi-task learning architecture for this problem?

RQ3: Can the proposed multi-task learning model leverage the shared signals between the correlated tasks?

RQ4: How does adding more tasks affect the model's performance in the primary tasks?

RQ5: Does an already trained multi-task model help in improving recall in a specific subset of data that could be of particular interest to the instructors? ¹

Our experiments show that automatic forum curation benefits from sharing signals between Confusion and Urgency detection, and our proposed multi-task learning architecture improves on the individual tasks by learning shared and mutually beneficial features between the tasks. We summarize our contributions as follows:

- We empirically explore the extent to which confusion and urgency detection are correlated using representative MOOC forum posts.
- We propose two multi-task learning architectures that share information between related tasks.
- Using representative forum posts, we empirically demonstrate that multi-task models improve over single-task models. Our proposed model is especially useful in detecting posts that express both confusion and urgency, which can be particularly relevant for forum curation.

2. RELATED WORK

As MOOCs have attracted millions of users worldwide, analyzing big data from online courses have become an indispensable means towards understanding students' learning patterns. In this regard, previous research has proposed models to predict dropout or success [7, 13, 14, 18], to measure the impact of social factors in attrition prediction [15], and to automatically curate discussion forums [2, 3, 4]. For example, Ramesh et al. [14] proposed a latent representation model which could be used to abstract student engagement types and to predict dropouts. Wang et al. [19] adopted a content analysis approach to investigate the relationship between students' cognitive behavior in MOOCs forums and their learning gains. Chaturvedi et al. [4] proposed chain-based models that incorporate meta-data along with course information and content of the posts to identify the posts

¹These are instances where posts are labeled as both Urgent and Confusion.

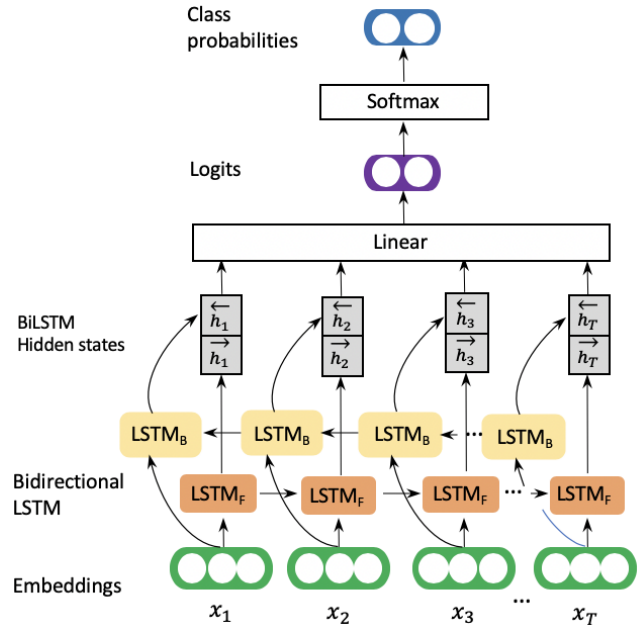


Figure 1: Single-Task Bidirectional LSTM Model.

that require instructor's attention. Chandrasekaran et al. [3] demonstrated the importance of prior knowledge about forum types in enhancing the predictive performance on the instructor's intervention task. Chandrasekaran et al. [2] proposed a supervised classifier which makes use of an automatic discourse parser for robust instructor intervention prediction.

Previous work has also focused on using behavioral and community-related cues to provide an insight into students' intentions, performances, and comprehension levels [21, 21]. Zeng et al. [22] and Agrawal et al. [1] investigated linguistic features along with structural features (e.g., the number of times a post has been read or the number of up-votes) to detect confusion. As identified by previous works [1, 22], one of the primary challenges in this area, is the lack of labeled instances and previous methods have explored the use of domain adaptation for addressing this challenge [23].

To address the problem of labeled data scarcity and leverage the relatedness between tasks, we propose to use multi-task learning which has been proven to perform well in many NLP tasks that include sequence labeling [5], text classification [10], machine translation [6]. For example, Liu et al. [9] proposed different architectures to control the information flow between shared or specific embedding and LSTM layers for text classification. However, multi-task learning has not been effectively explored for the online education domain. In this paper, we propose two multi-task frameworks to jointly learn related tasks (*confusion* and *urgency* detection) from the shared signals.

3. METHODOLOGY

We first define our task in Section 3.1 and in the following sections, we describe the Single-Task (ST), Shared-BiLSTM, and Specific-Shared Multi-Task (SSMT) models.

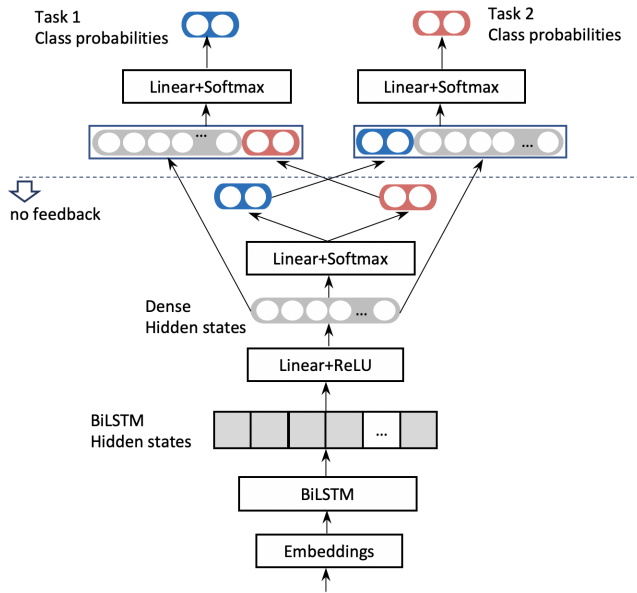


Figure 2: Shared-BiLSTM Model Architecture.

3.1 Problem Formulation

Our training dataset is $D = \{(X^i, Y^i)\}_{i=0}^N$, where X^i represents the i^{th} instance, and $Y^i = \{y_1^i, y_2^i, \dots, y_M^i\}$ denotes a set of M labels for the instance, one corresponding to each task². We assume that each task is a binary classification problem ($y_j^i \in \{0, 1\}$), but the proposed method can also work for multi-class classification tasks. In the following sections, we describe our different architectures.

3.2 Single-Task (ST)

We first create single-task models with identical architectures, to address the individual tasks of detecting confusion and urgency separately. The architecture is depicted in Figure 1. Given a forum post instance as a sequence of tokens $X^i = \{x_1, x_2, \dots, x_T\}$, and the class label Y^i , we first use an embedding layer to get the vector representation of each token x_t , followed by a BiLSTM layer and a linear layer with softmax activation to obtain class probabilities. The model is trained to minimize the cross-entropy loss for each task:

$$L = - \sum_{i=1}^N y^i \log(\hat{y}^i) \quad (1)$$

Where y and \hat{y} are the ground-truth and predicted labels (for a particular task) respectively.

3.3 Shared-BiLSTM

We now describe our first multi-task model that uses a shared BiLSTM encoder between different tasks to capture related information. The shared encoder has its architecture nearly identical to the single-task model except that it has an extra linear layer with ReLU activation between the BiLSTM and the Linear (with softmax) layers. Figure 2 shows the model architecture for two tasks, however; it can be easily extended for M tasks. Note that in this (and the next)

²In our case, we chose $M = 2$, where each label indicates if a post pertains to *confusion* and *urgency*.

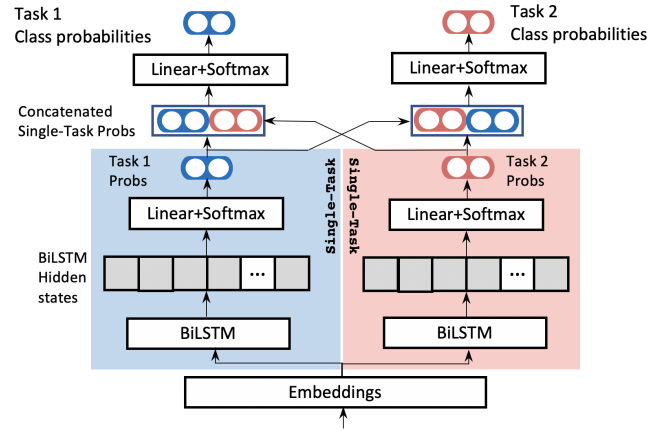


Figure 3: Specific-Shared Multi-Task (SSMT) Model Architecture

figure certain layers are collapsed into one single layer for simplicity. For instance, we depict Linear and Softmax as Linear+Softmax in Figure 2 and 3. We experimented with two main variations of this architecture: (1) Without feedback, and (2) With feedback. The first variation, without feedback, is the part of the model shown below the dotted line in Figure 2. The second variation, with feedback, has the class probabilities of each task concatenated with the dense hidden states (the entire Figure 2). Given the training pairs of a post sequence $X^i = \{x_1, x_2, \dots, x_T\}$, and the class label Y^i , the parameters of the model are updated to minimize total cross-entropy loss for the M tasks:

$$L_{total} = - \sum_{i=1}^N \sum_{j=1}^M y_j^i \log(\hat{y}_j^i) \quad (2)$$

3.4 Specific-Shared Multi-Task (SSMT)

We now describe our second multi-task model, Specific-Shared Multi-Task SSMT, that unlike the Shared BiLSTM model, first models task-specific characteristics and then shares information between the tasks. This model has *task-specific* components, with architectures identical to that of single-task models, to learn task-specific features (shown in highlighted parts of Figure 3)³. Thereafter, the model shares information across tasks by concatenating the predictions of the task-specific components followed by a fully connected layer (with softmax activation) to make predictions for the various tasks. Given the training pairs of post sequence $X^i = \{x_1, x_2, \dots, x_T\}$, and corresponding class labels Y^i , we first trained two separate single-task models, and used them to initialize the *task-specific* components of the multi-task network. We then trained the entire network to minimize the total cross-entropy loss defined in Equation 2. Note that during training, *task-specific* BiLSTM parameters get updated along with other model parameters.

4. EVALUATION

In this section, we evaluate the utility of the proposed multi-task models to address our primary tasks: Confusion and

³Like before, Figure 3 shows the architecture for two tasks, but can be easily extended for more tasks.

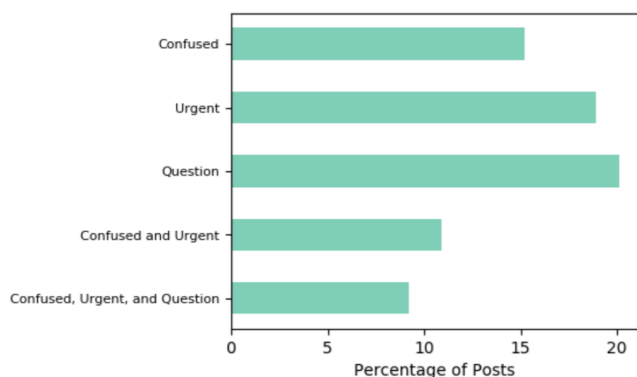


Figure 4: Label Distribution for the Stanford MOOC Posts Corpus

Urgency detection. Following previous works, we measure performance using Precision, Recall, and F1 scores of the positive class (*confusion* or *urgency*). This is because from the perspective of forum curation and helping students, positive class is more important than the negative class.

Dataset. We perform our experiments on the Stanford MOOC Posts Corpus [1]. The dataset contains 29,604 anonymized forum posts from 11 Stanford University public online classes spanning three broad domains: Humanities/Sciences, Medicine and Education. While this dataset has several labels, we primarily focus on two labels: Confusion and Urgency, labeled on a scale of 1 – 7. The confusion rating is based on the extent to which the post expresses confusion, such as an inability to understand some concept that is taught in the class. Similarly, the urgency rating is based on how urgent it is that the instructors respond to the post. Although these labels are on a scale of 1 – 7, following previous work [1], we convert these labels to binary values – posts with a score greater than 4 are categorized as *Confusion* (or *Urgency*), and those with a score equal or less than 4 as *Not Confusion* (or *Not Urgency*). Additionally, in some of our experiments, we use an additional label – *Question*, indicating whether the post was a question or not. Figure 4 shows the dataset’s label distribution. We can see that only 15.19% of posts are labeled as *Confusion*, which shows a severe class imbalance in this dataset. We use an 80 – 10 – 10 split for training, validation, and test data.

Training Details. For all our models, we initialized the embedding matrix with pre-trained 100-dimensional GloVe vectors [12]. We use a one-layer BiLSTM network with 80 hidden units. We experimented with using more layers and hidden units. However, that led to over-fitting possibly because of the relatively smaller size of the dataset. We applied dropout [17] of rate 0.2 between the BiLSTM hidden layers and the output layers for regularization, and did not fine-tune the word embeddings during training to avoid over-fitting. Finally, we optimized using the Adam optimizer [8], with a learning rate of 0.001.

Correlation Analysis. We performed inter-label correlation analysis prior to our main experiments. First, we visualize the relationship between Confusion and Urgency (considering the original (1 – 7) Likert scale) in the boxen

plot shown in Figure 5. We can see that there can be disagreement between confusion and urgency labels especially around the threshold rating of 4. For example, there are several posts with confusion rating of 4.5 which would be labeled as *Confusion* but not *Urgency* (because their urgency ratings are less than 4). However, we observe a relatively high correlation between the two tasks for the most part.

Next, we also analyze the Spearman correlation between *confusion* and *urgency* (Table 1). We consider both original as well as the binary labels based on the threshold described earlier. We observe a moderate correlation between Confusion and Urgency (0.570). We also report correlations of these labels with respect to Question to explore whether it can be additionally used in the multi-task setup to improve the performance of Confusion and Urgency detection (the two primary tasks we are interested in). We also find that using binary labels increases the inter-label correlation for all cases. Note that inter-label correlation suggests but does not guarantee or quantify improvement in predictive performance with multi-task learning. Hence, in the following section, we design a new experiment where we consider three single-task models (confusion, urgency, and question) and explore the utility of each to predict Confusion and Urgency (RQ1). We then conduct other experiments to further investigate the utility of multi-tasking for these problems.

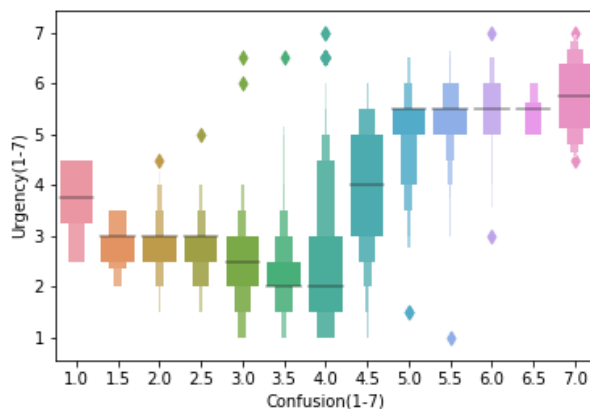


Figure 5: Inter-label correlation distribution between ordinal Confusion and Urgency label; the Spearman correlation value is 0.481.

4.1 Experimental Results

In our experiments, we implemented a single-task architecture mentioned in Section 3.2 to create models for each of the three tasks by training them on labeled data from the respective tasks: Single-Task Confusion detection (ST-C), Single-Task Urgency detection (ST-U), and Single-Task Question detection (ST-Q). These form our baselines. We follow a similar naming convention for the Shared-Specific Multi-Task model. For example, we refer to the Shared-Specific Multi-Task model to predict confusion and urgency together as SSMT-CU.

As a preliminary experiment, we compare the performances of our neural Single-Task models with Logistic Regression (LR) using Bag-of-Words and tf-idf features. Comparing the results in Table 2 with those in Rows 1 and 4 of Table 3, we

Labels	Confusion(1-7)	Confusion(1/0)	Urgency(1-7)	Urgency(1/0)	Question(1/0)
Confusion(1-7)	1.0	0.722	0.481	0.545	0.510
Confusion(1/0)	0.722	1.0	0.603	0.570	0.567
Urgency(1-7)	0.481	0.603	1.0	0.852	0.671
Urgency(1/0)	0.545	0.570	0.852	1.0	0.690
Question(1/0)	0.510	0.567	0.671	0.690	1.0

Table 1: Spearman correlation between all labels

Model	Task predicted	F1	Precision	Recall
LR-C+BOW	Confusion	0.45	0.56	0.38
LR-C+tf-idf	Confusion	0.38	0.68	0.27
LR-U+BOW	Urgency	0.61	0.67	0.57
LR-U+tf-idf	Urgency	0.59	0.76	0.48

Table 2: Performance evaluation of single-task models with Logistic Regression as baseline

can see that ST-U and ST-C outperform Logistic Regression based models on both the tasks. So, henceforth we use our neural models for all single task experiments.

RQ1: To what extent are different tasks in this domain correlated?

The goal of our first experiment is to find out if the tasks are correlated enough that model trained on one task can yield reasonable predictive performance on the other task. This would indicate if multi-tasking can help for jointly learning these tasks. For this purpose, we first evaluate ST-C, ST-U, and ST-Q on the task of confusion detection. Even though ST-U and ST-Q were not trained on this label (confusion), we posit that since the tasks of urgency and question detection are correlated with that of confusion detection, these models could have learned signals commonly shared with the confusion detection task. We perform a similar experiment to find correlations with urgency detection. All results are reported in Table 3.

The experiment indicates that the strongest correlation exists between the primary tasks: Confusion and Urgency detection. When used to predict the confusion label, ST-U obtains an F1 score of 0.47, which is only slightly lower than that obtained by ST-C (0.50). Similarly, ST-C performs relatively well in the urgency detection task suggesting that ST-U and ST-C have learned mutually beneficial signals, and can be used in a multi-task setup.

On the other hand, according to row 3 of Table 3, ST-Q has not learn enough mutually beneficial signals for the confusion detection task, suggesting that confusion and urgency are more useful for each other than question.

RQ2: What is an effective multi-task learning architecture for this problem?

We experimented with various versions of the two multi-tasking architectures proposed in Section 3. Here, we summarize these architectures and their performances.

For the Shared-BiLSTM model, we consider a variation without feedback (see Section 3.3) and three others with feedback. For the variations with feedback, we experimented

Model	Task predicted	F1	Precision	Recall
ST-C	Confusion	0.50	0.68	0.40
ST-U	Confusion	0.47	0.46	0.48
ST-Q	Confusion	0.32	0.39	0.27
ST-U	Urgency	0.67	0.72	0.62
ST-C	Urgency	0.44	0.67	0.33
ST-Q	Urgency	0.44	0.60	0.47

Table 3: Performance evaluation of single-task models when used to predict *Confusion* or *Urgency*

with (1) Initializing the entire network randomly, (2) Pre-training and then freezing the shared encoder, and (3) Pre-training the shared encoder but further tuning the entire model to minimize total loss. Together these make up a total of 4 variations of the Shared-BiLSTM model. The performances are reported in the top half of Table 4. We can see that the variations which performed the best are the one that includes feedback with random initialization and the one with feedback, pre-training and freezing.

We also experimented three variations of SSMT: (1) Adding an extra Linear layer with ReLU activation between BiLSTM and final Linear Layers, (2) Including single-task losses in Equation 2 when fine-tuning the entire network, and (3) The model described in Section 3.4 without any changes. The results are summarised in lower half of Table 4. We can see that the model without any changes (as described in Section 3.4) outperforms its other two variations as well as all variations of the Shared-BiLSTM architecture. For the rest of our experiments we use SSMT as our final multi-task setup and we discuss its performance in the rest of the research questions.

RQ3: Can the Specific-Shared Multi-Task model leverage the shared signals between the correlated tasks?

We evaluate our Specific-Shared Multi-Task model for predicting Confusion and Urgency (SSMT-CU). Table 5 shows that SSMT-CU outperforms both ST-C and ST-U on the two primary tasks. Comparing Rows 1 and 3, there is an increase in F1 score for the confusion detection task from 0.50 to 0.56. Comparing Rows 4 and 6 shows that we also obtain a boost in F1 score of the urgency detection task from 0.67 to 0.69. These results are statistically significant ($p < 0.001$) [11], and indicate that SSMT-CU has learned the shared signals between the two tasks. Also, we see that urgency has helped to identify confusion more than vice versa. This can also be observed in Table 3: the drop in performance when using ST-U instead of ST-C for confusion detection was much smaller than the drop resulting from using ST-C instead of ST-U for urgency detection. This also hints that urgency signals are more useful for confusion detection

	CONFUSION			URGENCY		
	F1	Precision	Recall	F1	Precision	Recall
Shared-BiLSTM (w/o fb)	0.50	0.64	0.41	0.66	0.65	0.68
Shared-BiLSTM (+fbrandom-initialization)	0.53	0.67	0.44	0.63	0.70	0.57
Shared-BiLSTM (+fb+pre-training+freeze)	0.53	0.67	0.43	0.67	0.69	0.66
Shared-BiLSTM (+fb+pre-training+tune)	0.48	0.72	0.35	0.63	0.72	0.57
Specific-Shared Multi-Task (+dense)	0.52	0.62	0.44	0.68	0.66	0.70
Specific-Shared Multi-Task (+st-losses)	0.52	0.66	0.42	0.68	0.70	0.67
Specific-Shared Multi-Task	0.56	0.66	0.49	0.69	0.70	0.69

Table 4: Results of different variations of our two multi-task architectures. We indicate feedback with “fb”, and single-task with “st”. Bold fonts denote best performances among top and bottom halves of the table.

Model	Task predicted	F1	Precision	Recall
ST-C	Confusion	0.50	0.68	0.40
SSMT-CUQ	Confusion	0.52	0.71	0.41
SSMT-CU	Confusion	0.56	0.66	0.49
ST-U	Urgency	0.67	0.72	0.62
SSMT-CUQ	Urgency	0.69	0.71	0.67
SSMT-CU	Urgency	0.69	0.70	0.69

Table 5: Performance evaluation of single-task and multi-task models; MT-CU and SSMT-CUQ outperform ST-C and ST-U in the primary tasks.

than confusion signals for urgency detection.

RQ4: How does adding more tasks affect the model’s performance in the primary tasks?

To investigate if adding the task of Question Detection can supplement the primary tasks, we introduce the SSMT-CUQ model and compare it with the existing models. Comparing Row 1 with 2, and 4 with 5 in Table 5, we find that SSMT-CUQ has a better F1 score than both ST-C and ST-U. This shows that adding an extra task still yields better performance than single-task models for the primary tasks.

To evaluate whether it further enhanced the SSMT-CU model, we compare Rows 2 with 3 and 5 with 6. SSMT-CU obtains a higher F1 score (0.56) than SSMT-CUQ (0.52) for the confusion detection task. We attribute the drop in performance of SSMT-CUQ for the confusion task to the relatively weaker correlation between the question detection and confusion detection tasks (also observed in our earlier experiment when comparing Rows 1 and 3 of Table 3). The introduction of question detection task might have introduced more noise and weakened the shared signals of confusion and urgency.

On the other hand, SSMT-CUQ and SSMT-CU have identical F1 scores (0.69) on the urgency detection task (Rows 5 and 6 of Table 5). Despite question detection being as useful for urgency detection as confusion detection (shown in Table 3), SSMT-CUQ did not improve over SSMT-CU because it might have received similar signals from both confusion detection and question detection tasks.

RQ5: Does an already trained multi-task model help improving recall in an specific subset of data that could be of particular interest to the instructors?

We now turn our attention to a specific subset of our dataset – posts labeled as both urgent as well as expressing confu-

Model	Confusion Recall	Urgency Recall
SSMT-CU	0.59	0.70
ST-C	0.49	-
ST-U	-	0.59

Table 6: Performance evaluation for the subset of confused and urgent posts

sion – for their potential to impact learner satisfaction ⁴. In this experiment, the models are not trained on this subset. Instead, we analyze the performance of the (already trained) multi-task model on this subset. Since all the posts in this subset are labeled as *Confusion* and *Urgency*, any model will have a precision of 1 leading to a less informative F1 score. So, in this experiment, we focus on Recall values. Table 6 shows that the Specific-Shared Multi-Task model significantly outperforms the single-task models in the subset for both confusion and urgency ($p < 0.001$).

These results indicate that by leveraging correlated tasks in the multi-task setting, the SSMT model has learned hidden abstractions which help it to outperform single-task models trained solely on confusion or urgency not just in general, but also in the more important subset of the data.

5. CONCLUSION

In this paper, we hypothesize that inter-label correlation or co-occurrence counts suggest but do not guarantee or quantify improvement in predictive performance with multi-task learning. This prompts us to design several experiments to explore the benefits of multi-task learning for confusion and urgency detection in MOOCs forums. We propose the SSMT model, a multi-task learning framework, to facilitate forum curation. We demonstrate that our proposed model outperforms single-task models consistently across both tasks. The multi-task framework takes advantage of the shared signals to yield not only superior performance in general, but also in the subset of the data that is most important for curation: posts that express both confusion and urgency. Future work can extend multi-task learning to explore its generalization performance across various course offerings. More specifically, it can investigate whether a multi-task learner trained on one course, can be effectively used for prediction in other related courses. In this regard, multi-task-based unsupervised domain adaptation can be applied to jointly learn the source and target course classifiers.

⁴We created this subset from test set of our data.

References

- [1] Akshay Agrawal, Jagadish Venkatraman, Shane Leonard, and Andreas Paepcke. 2015. Youedu: Addressing confusion in MOOC discussion forums by recommending instructional video clips. In *Proceedings of the 8th International Conference on Educational Data Mining, EDM 2015*, pages 297–304.
- [2] Muthu Kumar Chandrasekaran, Carrie Demmans Epp, Min-Yen Kan, and Diane J. Litman. 2017. Using discourse signals for robust instructor intervention prediction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, pages 3415–3421.
- [3] Muthu Kumar Chandrasekaran, Min-Yen Kan, Bernard C. Y. Tan, and Kiruthika Ragupathi. 2015. Learning instructor intervention from MOOC forums: Early results and issues. In *Proceedings of the 8th International Conference on Educational Data Mining, EDM 2015*, pages 218–225.
- [4] Snigdha Chaturvedi, Dan Goldwasser, and Hal Daumé III. 2014. Predicting instructor’s intervention in MOOC forums. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1501–1511.
- [5] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008)*, pages 160–167.
- [6] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (IJNLP)*, pages 1723–1732.
- [7] Josh Gardner and Christopher Brooks. 2018. <https://doi.org/10.1007/s11257-018-9203-z> Student success prediction in MOOCs. *User Modeling and User-Adapted Interaction*, 28(2):127–203.
- [8] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*.
- [9] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, page 2873–2879.
- [10] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. <https://doi.org/10.18653/v1/P17-1001> Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 1–10.
- [11] Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley New York.
- [12] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. <https://doi.org/10.3115/v1/D14-1162> Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.
- [13] Antoine Pigeau, Olivier Aubert, and Yannick Prié. 2019. Success prediction in MOOCs: A case study. In *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019*, pages 390–395.
- [14] Arti Ramesh, Dan Goldwasser, Bert Huang, Hal Daumé, and Lise Getoor. 2014. Learning latent engagement patterns of students in online courses. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, page 1272–1278.
- [15] Carolyn Penstein Rosé, Ryan Carlson, Diyi Yang, Miaomiao Wen, Lauren Resnick, Pam Goldman, and Jennifer Sherer. 2014. <https://doi.org/10.1145/2556325.2567879> Social factors that contribute to attrition in MOOCs. In *Proceedings of the First ACM Conference on Learning @ Scale Conference*, page 197–198.
- [16] Julia E Seaman, I Elaine Allen, and Jeff Seaman. 2018. Grade increase: Tracking distance education in the united states. *Babson Survey Research Group*.
- [17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- [18] Feng Wang and Li Chen. 2016. A nonlinear state space model for identifying at-risk students in open online courses. In *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016*, pages 527–532.
- [19] Xu Wang, Diyi Yang, Miaomiao Wen, Kenneth R. Koedinger, and Carolyn Penstein Rosé. 2015. Investigating how student’s cognitive behavior in MOOC discussion forum affect learning gains. In *Proceedings of the 8th International Conference on Educational Data Mining, EDM 2015*, pages 226–233.
- [20] Xiacong Wei, Hongfei Lin, Liang Yang, and Yuhai Yu. 2017. A convolution-lstm-based deep neural network for cross-domain MOOC forum post classification. *Information*, 8(3):92.
- [21] Diyi Yang, Miaomiao Wen, Iris Howley, Robert Kraut, and Carolyn Rose. 2015. Exploring the effect of confusion in discussion forums of massive open online courses. In *Proceedings of the second (2015) ACM conference on learning@ scale*, pages 121–130.
- [22] Ziheng Zeng, Snigdha Chaturvedi, and Suma Bhat. 2017. Learner affect through the looking glass: Characterization and detection of confusion in online courses. In *Proceedings of the 10th International Conference on Educational Data Mining, EDM 2017*, pages 272–277.

- [23] Ziheng Zeng, Snigdha Chaturvedi, Suma Bhat, and Dan Roth. 2019. <https://doi.org/10.1145/3303772.3303810> DiAd: Domain adaptation for learning at scale. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge, LAK 2019*, pages 185–194.
- [24] Yu Zhang, Ying Wei, and Qiang Yang. 2018. Learning to multitask. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5771–5782.
- [25] Yu Zhang and Qiang Yang. 2017. <http://arxiv.org/abs/1707.08114> A survey on multi-task learning. *CoRR*, abs/1707.08114.

CSCLRec: Personalized Recommendation of Forum Posts to Support Socio-collaborative Learning

Zhaorui Chen and Carrie DEMMANS EPP

EdTeKLA Research Group, Dept. of Computing Science, University of Alberta
{zhaorui, cdemmansepp}@ualberta.ca

ABSTRACT

Discussion forums are used to support socio-collaborative learning processes among students in online courses. However, complex forum structures and lengthy discourse require that students spend their limited time searching and filtering through posts to find those that are relevant to them rather than spending that time engaged in other meaningful learning activities (i.e., discussion). Moreover, existing adaptive systems do not accommodate individual learner needs in these contexts. In this work, we propose a multi-relational graph-based recommendation approach that mines student interaction logs to address the above problems within discussion-based socio-collaborative online courses. To account for the social aspects of learning, our approach incorporates learner modeling, social network analysis, and natural language processing techniques; it offers tailored recommendations of forum posts for learners with different types of interaction behaviors. In our experiments with small online courses, our approach outperformed competitor approaches in terms of recommendation precision while meeting expectations with respect to diversity and novelty. The results illustrate the proposed algorithm's effectiveness in predicting student preferences, suggesting its potential to increase student participation in discussion-related learning activities.

Keywords

Recommender systems, Discussion forums, Computer-supported collaborative learning, Online learning.

1. INTRODUCTION

Asynchronous online discussion forums are widely used to support online courses in higher education [5, 23, 25]. In these forums, many instructors post discussion topics and encourage students to expand so that knowledge can be co-created and developed through progressive discussion. In such socio-collaborative learning contexts, students' active participation and production of learning resources is essential, as less discussion could result in less sharable knowledge and thus less learning within a course [37, 81]. However, forums' complex thread structure and information-heavy posts tend to have a negative impact on student engagement, because much of their time is spent locating relevant forum posts, rather than focusing on core tasks such as debating, reflecting, and learning from each other [1, 37]. To alleviate this type of information overload problem, deploying recommender systems to recommend posts of interest or content generated by others could be beneficial.

Many recommender systems have been used to support learning across varied domains and contexts. For example, data mining approaches were used to suggest course improvements in learning management systems [33], and a workplace learning support system paired users with knowledgeable peers to enable knowledge sharing processes [8]. More recently, other systems have recommended courses to university students [7, 29, 65].

While these examples show the prior success of recommender systems in educational contexts, few have solved the problem of recommending socio-collaborative learning materials in discussion forums for smaller online courses. To fill this gap, we present a novel graph-based recommender system approach. This approach mines learner interaction data using both modelled learner types and natural language processing techniques that were specifically designed for this application domain of smaller discussion-based socio-collaborative learning environments. In our research, we posed the following question: *How do traditional recommender algorithms and those that incorporate principles from socio-collaborative learning perform when suggesting posts in small online socio-collaborative learning contexts?*

2. Related Work

2.1 Socio-collaborative Learning

Socio-collaborative learning, also known as collaborative learning, refers to a class of learning methods in which learners cooperate in a group, relying on each other, being responsible for each other, and accomplishing a common task together [75]. This approach can be traced back to Vygotsky who pointed out that those who are more able can help others perform better [83]. Piaget claimed that the cognitive conflicts generated during social interaction could help the learner reflect on their original point of view, thus enhancing their understanding [44]. Subsequently, collaborative learning has become a widely used pedagogical theory that is also a target of many online learning environments, where it is called computer-supported collaborative learning (CSCL).

CSCL often occurs through online discussion-based forums [17, 26, 39] where information is transmitted through posts to enable knowledge sharing or co-construction among learners. The systems and mechanisms used to support CSCL are grounded in theories such as knowledge building: a specific knowledge co-construction process that emphasizes the creation of ideas through discussion [70, 71]. Many of the proposed knowledge building principles (e.g., diverse and improvable ideas or symmetric knowledge advancement [69]) provide theoretical support for our research.

2.2 Recommenders for Educational Forums

Most work has focused on supporting question and answer (Q&A) forums in university courses [34], MOOCs [49, 53, 87] or other online educational platforms [41, 80] when recommending forum posts. These systems typically aim to reduce the number of unanswered questions by recommending 1) unanswered questions

Zhaorui Chen and Carrie Demmans Epp "CSCLRec: Personalized Recommendation of Forum Posts to Support Socio-collaborative Learning" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 364 - 373

to students who are able to answer them, and 2) similar questions that have already been answered to users who are about to ask one.

Using a similar recommender system design in smaller-scale socio-collaborative settings is inappropriate because the contexts differ in terms of size and pedagogical purpose. In contrast to MOOCs, these contexts suffer from both a lack of data and the cold start problem. Different from Q&A forums, developing knowledge sharing processes in discussion forums requires the algorithm support increased connectivity among users to facilitate communication [45]. It is also necessary to include posts containing diverse and novel ideas from students who express different points of view so that they might learn from each other [70].

Few studies have investigated how to deal with these challenges. Those that have depend on a priori domain knowledge (e.g., rules [2] or ontologies [18]) which is time-consuming to obtain and has limited generalizability to unseen cases [43]. Given increases in online course delivery and a desire to support students' socio-emotional development and collaborative learning [3, 24], we need approaches that can be used in the absence of domain expertise.

Many have also argued that CSCL personalization technologies should consider the social [42, 66] and other needs of learners [12, 51, 68, 79]. One study investigated learners' knowledge sharing behaviors in closely-knit communities to generate tailored notifications [45]. The notifications aimed to foster knowledge sharing processes within a learning community composed of different learner types. To extend this idea to the context of a forum post recommender system, we set out to develop recommendation algorithms that also consider learner socio-behavioral patterns and created customized strategies for each behavior pattern.

3. Recommender Algorithm: CSCLRec¹

The target users of this system are students or learners. We will use these terms interchangeably. Our proposed algorithm, CSCLRec, relies on 3 types of data that are available in any educational forum: user interactions with forum posts which we call user-to-post (U2P) interactions; communication between users, such as reading, that we call user-to-user (U2U) interactions; and the textual content of forum posts. Using this data, it recommends posts to learners.

CSCLRec has four modules (see Figure 1): a personalized PageRank graph, a learner interaction profiler that analyzes U2U interactions, a content analyzer, and a post filtering module.

3.1 Personalized PageRank Graph

The core of the system is a modified personalized PageRank (PPR) graph [35]. As shown in Figure 2, the PPR has nodes for users, posts, and hypernyms. A hypernym is a superordinate word whose semantic meaning includes a set of other words. For example, "flower" is the hypernym for "rose" or "daisy". Multiple types of relationships including U2P interactions, inter-user relationships, and posts' relationship with hypernyms are computed by other modules and represented as edges in the graph. The weight of user-to-post edges in the graph is biased by a temporal decay rate. Edges representing U2P interactions in the past have lower weights so that the algorithm can focus on the user's recent interests.

We refer to the user who is receiving the recommendations as the active user. To recommend posts to an active user, the algorithm performs a random walk starting from their user node and its

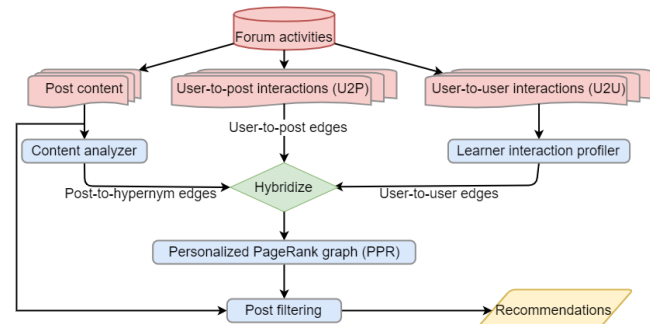


Figure 1. Overall workflow of CSCLRec

connected post nodes. When sufficient iterations have been completed, the nodes' probabilities of being visited by the random walk agent will converge to a steady state. Posts with the highest probability of being visited are presented as the recommendations. We used power iteration [60] to approximate the stationary probabilities and avoid poor computational performance.

3.2 Learner Interaction Profiler

The learner interaction profiler uses a bidirectional social network graph, which consists of different types of U2U interactions (e.g., replies and reads). Each user is a node in this graph and the interactions among users are edges. In Figure 3, the thin grey link from user U1 to user U2 indicates that U1 has read U2's post.

Students who have many interactions with the active user are their peer learners. The rich interaction history, whether in discussion or debate, indicates the active user's interest in interacting with those peers. This group of users share many outward edges with the active user in the social network graph. The inclusion threshold for number of edges required between users is set via grid search. As a result, the module generates links connecting the active user to those peer learners (the green edges in Figure 2) in the PPR graph.

The analyses over the graph also output a participation level (i.e., number of outgoing edges of reply, like, and link types from its user node) and a degree of centrality (i.e., the in-degree of a node) for each student. The more frequently other students interact with the active user's posts, the more they can increase the active user's degree of centrality. The participation level indicates the extent to which the student is actively engaging in the discussion. We used the two measures to identify four types of learners (new user, listener, single-pass user, and peripheral user) that may need differentiated recommendation strategies. These user types are identified using simple heuristics based on the literature.

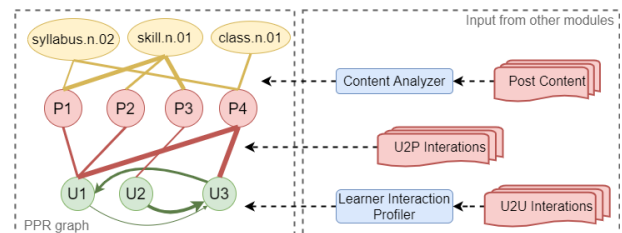


Figure 2. The modified PPR graph has 3 node types (user - green, post - red, and hypernym - yellow) and 3 edge types (user-to-user - green, user-to-post - red, and post-to-hypernym - yellow). Edges without arrows are bidirectional and edge width indicates number of occurrences.

¹ Code is available at <https://github.com/EdTeKLA/CSCLRec>

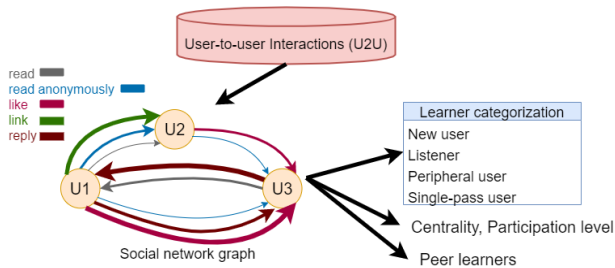


Figure 3. The workflow of the learner interaction profiler. Edge widths in the graph indicate the number of interactions.

New users are learners who have just joined the discussion. They have not created any resources nor do they have any other logged interactions. Consequently, these users are subject to the cold-start problem, which makes it difficult for the algorithms to provide suggestions because of the lack of data [10]. New users may experience greater information overload because they face many posts at once and may need tailored recommendations to help them filter information, identify their interests early, and contribute their own voices. To prevent narrow recommendations, new users are connected to every other user in the PPR graph.

Listeners read many posts but rarely post themselves [86]. The knowledge building principle of collective responsibility and symmetric knowledge advancement suggests that encouraging posting is critical to fostering activity and promoting knowledge co-construction [69]. Listeners are identified as those who have not created posts. To reduce the number of persistent listeners, we adopt the same recommendation strategy as that employed for new users since exposing these learners to different topics may increase the possibility of their expressing opinions [46].

Peripheral users are those whose centrality score is decreasing due to lost interactivity in their readership. The module aims to recover peripheral users and listeners to promote the knowledge-sharing process [47, 48] and prevent the loss of these readers' activity and interest. The learner profiler monitors the number of interactive readers for each learner: those who reply, like, or link. When the profiler detects the user's interactive reader count has dropped by half from one week to the next, that user is marked as a peripheral user. This value was tuned during the evaluation. The algorithm takes note of the lost readers and introduces connections between the peripheral user and the lost readers in the PPR graph to strengthen their connections.

Single-pass users only read new posts and ignore older posts [38]. Their widespread presence undermines socio-collaborative learning approaches because these learning processes require topics to be progressively discussed and deepened [69]. To alleviate this behavior, some have suggested encouraging students to revisit earlier posts [38]. Inspired by this idea, the learner interaction profiler identifies students who have only read posts from the previous week. For example, those who have not read posts created before week 7 are marked as single-pass users in week 8. The modified PPR graph decreases the temporal decay exerted on older posts for single-pass users so earlier posts are down-weighted less, increasing the likelihood of their recommendation to these learners.

3.3 Content Analyzer

Forum posts are hierarchically structured. Posts on the same topic have similar interaction records because users are accustomed to browsing the entire topic structure when reading a post. Therefore, algorithms based on interaction records (i.e., collaborative filtering,

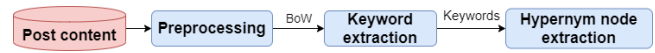


Figure 4. The workflow of the content analyzer module

ordinary personalized PageRank bipartite graph) may only recommend posts that are locationally similar to those that users often interact with. Consequently, students may lose the opportunity to read posts that match their current interests because they are located elsewhere. These algorithms also bias towards post popularity [77] causing the “long-tail” problem: unpopular posts are not considered for recommendation [21, 62], which could decrease student exposure to diverse perspectives. To overcome these challenges, the content analyzer module applies natural language processing (NLP) techniques to the content of forum posts and enables links to be created between posts based on the concepts discussed rather than user interactions (as shown in Figure 2). Its workflow is shown in Figure 4. The preprocessing stage removes all html mark-up and punctuation. It also tokenizes sentences into individual words. Lemmas are extracted for nouns and verbs, and stop words are removed. To protect user privacy, person names, usernames, web URLs, and email addresses are also removed. Each post is then organized as a bag-of-words (BoW).

TF-IDF was chosen for keyword extraction following a preliminary evaluation that compared several potential methods (i.e., RAKE [67], TextRank [54]) on independent data from the same system. TF-IDF scores are computed for each lemma to choose keywords that best differentiate the current post from others. The keywords with the top 1/5 TF-IDF scores are used to represent the post.

The extracted keywords are used to measure thematic similarity across posts. Instead of matching keywords using text similarity approaches (i.e., sentence embeddings or topic distribution vectors in vector space models), we consider two posts thematically similar provided they mentioned similar concepts regardless of student opinion towards a topic. The tools used to measure similarity included the WordNet semantic network and its collection of hypernyms [55]. We query each post in WordNet and use Lesk [50] to disambiguate hypernyms. The hypernyms are added as nodes in the PPR graph - see the yellow nodes in Figure 2. When a post contains a keyword that belongs to this hypernym, a link from the post node to the hypernym node is constructed. As a result, posts that share more concepts will share more hypernym nodes.

3.4 Post Filtering

This module analyzes, sorts, filters, and re-ranks the results produced by the recommender which may otherwise include less-informative posts that will not advance student knowledge. Posts like, “Thank you for the clarification, [name]” may be output by the algorithm if this filtering is not performed. The post filtering module refines the recommendations using two filters: one extracts verb and noun phrases as trigram models and excludes posts with fewer than 3 phrases, and the other compares post content with the Academic Word List (AWL) [20]. Posts with fewer than 3 AWL words are removed.

4. METHODS

We evaluated the performance of CSCLRec, its precursor, and other widely-used algorithms using a similar protocol to that advised by recommender system researchers [28, 73]. In each week, we recommend 10 posts to each user. Posts were selected from a candidate list consisting of those the active user has not yet read and all posts created by others in the current (evaluation) week. We hide this user's activities from the evaluation week and use forum

Table 1. Student and instructor interactions through the course forum as a raw count (#) or $M(SD)$.

Course	Weeks (#)	Students (#)	Instructors (#)	Posts (#)	Interacted posts/ student	Interactions/ student	Reads/ student	Likes/ student	Links/ student
LA	13	26	1	1751	1176 (550.18)	1628 (1010.30)	1314 (719.23)	76 (61.16)	1.19 (2.98)
LB	13	19	4	809	358 (245.96)	441 (298.75)	365 (247.00)	21 (24.96)	0.05 (0.23)
LC	13	30	4	2090	1212 (686.41)	1373 (732.37)	1226 (698.94)	29 (23.88)	10.06 (15.20)
SA	6	23	1	627	362 (219.05)	505 (417.60)	405 (290.29)	15 (19.45)	0.26 (1.25)
SB	6	24	1	1142	616 (269.83)	731 (281.87)	635 (270.13)	8 (9.97)	0.25 (1.03)
SC	6	20	1	869	507 (223.65)	631 (269.11)	521 (223.42)	44 (44.63)	0.55 (1.57)

activities from prior weeks to train the recommenders. We start from week 2 since there are no learner posts prior to week 1.

4.1 Dataset

The evaluation used historical data from six postgraduate courses offered through an asynchronous discussion platform (PeppeR) at the University of Toronto. PeppeR provides a collaborative learning space to discuss and share ideas, making this system an ideal testbed to evaluate the proposed recommender system.

Archival data from fully online courses were used. Of the six test courses, three were regular-length courses (13-weeks long) and the others were short courses (6-weeks long). User activity statistics for each course are summarized in Table 1. The data includes forum posts and all kinds of user interactions with posts (i.e., posting, replying to others' posts, inserting hyperlinks to other posts, liking posts, and reading posts). The interactions were categorized into 7 types: create, reply, like, link, revisit, read, and anonymously read.

The large variability in student interactions (see Table 1) is consistent with the different types of users identified: Some had many forum activities, while others seldom interacted with posts. This suggests the necessity of distinguishing different learner types and employing user-specific recommendation strategies.

4.2 Recommender Algorithms

CSCLRec's performance was compared against that of 7 other recommenders. Due to the limited number of students, the diversity of student interaction behaviors, and the inter-dependence of time-series data, random cross validation was not appropriate. We tuned the hyperparameters using last block validation [9]: For each weekly evaluation, we used the prior week to validate the current weeks' recommendations. We used grid-search on the hyperparameters and trained the recommenders using data from before the validation week. Testing used data from the validation week. Using precision, the best performing hyperparameters were selected to build the recommenders for subsequent evaluations. For CSCLRec, we tuned temporal decay and the number of peer learners. All PPR-based algorithms had their damping factor tuned.

The algorithms we tested CSCLRec against are listed below. Hyper-parameter values are reported in the repository¹.

- Co-occurrence graph-based personalized PageRank (**CoPPR**) is another original method we developed. It uses the same learner profiler and post filtering modules as CSCLRec. Different from CSCLRec, Co-PPR uses the extracted keywords as nodes. Two keyword nodes are connected if they co-occurred at least once in a post. A post is connected to a keyword node if that post contains the keyword at least once. Edge weights are determined using the posts' keyword occurrence count. CoPPR helps identify the contribution of the content analyzer to CSCLRec. We tuned temporal decay, the damping factor, and number of peer learners.

- Personalized PageRank (**PPR**) is a widely used graph-based recommender [15, 58]. It uses a bipartite graph with user-to-post interactions as the only input.
- Matrix factorization collaborative filtering (**MCF**) represents a family of model-based collaborative filtering algorithms, which are commonly used in educational recommender systems [27, 78]. We used the version proposed by Hu and colleagues [40]. We tuned its confidence factor which specifies the negative weight attributed to unseen interactions.
- Keyword-based content-based recommender system (**KCB**) is frequently used to personalize discussion forums [4] and help-seeking platforms [52]. KCB relies on latent semantic indexing to create vectors from posts. Users are represented as the average of the post vectors they have interacted with before. It recommends candidate posts which are nearest to the active user in the vector space. The hyperparameters include the dimension of post vectors and the ratio of content words as the keywords (i.e., 1/7 of the content words are treated as keywords).
- Sentence embedding-based content-based recommender (**SCB**) relies on the semantics of post content [16].
- Popularity-based recommender (**PPL**) recommends popular posts. Every user receives the same recommendations. This unpersonalized algorithm is used as a baseline.
- The random recommender (**RND**) randomly draws posts from the candidate list. This algorithm is also used as a baseline.

We did not test all well-known recommendation algorithms as some structural aspects and requirements of the algorithms make them a poor fit given the nature of our dataset. For example, deep learning-based methods (i.e., autoencoders) are data-hungry and can easily overfit due to the size of our dataset [88].

4.3 Measures

Since accuracy is insufficient for determining the quality of educational recommender systems [30], we measured 3 dimensions of performance: accuracy, diversity, and novelty.

For accuracy, we report both Precision at K (**P@K**) and Recall at K (**R@K**), where k is the number of recommendations. The R@K measure is affected by the number of available relevant items [73] so we report the maximum (max) R@10 to aid interpretation. Max R@10 is the average of the largest possible R@10 in each user's recommendations. We adopted the commonly used intra-list diversity (**ILD**) indicator which measures the average pairwise distance between recommended items [14, 76]. We used pre-trained Universal Sentence Encoder [16] embeddings to represent the posts and the cosine distance to compute ILDs. The mean inverse user frequency (**MIUF**) indicator is used to measure recommendation novelty [11]. The fewer people who have interacted with the post, the higher the novelty and IUF of that post. To reflect the consistency of algorithm performance, we report the

Table 2. Summary of evaluation results as $M(SD)$

Algorithm	Long courses (LA, LB, LC)				Short courses (SA, SB, SC)			
	P@10	R@10	ILD	MIUF	P@10	R@10	ILD	MIUF
CSCLRec	0.729 (0.319)	0.219 (0.305)	0.274 (0.125)	0.612 (0.360)	0.751 (0.310)	0.188 (0.254)	0.191 (0.059)	0.482 (0.140)
CoPPR	0.718 (0.324)	0.221 (0.304)	0.222 (0.110)	0.638 (0.380)	0.731 (0.315)	0.177 (0.243)	0.156 (0.048)	0.502 (0.137)
PPR	0.537 (0.408)	0.178 (0.310)	0.390 (0.162)	0.466 (0.251)	0.566 (0.383)	0.142 (0.248)	0.244 (0.106)	0.407 (0.144)
MCF	0.484 (0.391)	0.180 (0.313)	0.449 (0.192)	0.837 (0.532)	0.449 (0.406)	0.130 (0.265)	0.357 (0.151)	0.801 (0.485)
SCB	0.294 (0.355)	0.158 (0.313)	0.075 (0.047)	1.216 (0.453)	0.400 (0.378)	0.117 (0.247)	0.079 (0.019)	0.927 (0.251)
KCB	0.289 (0.359)	0.150 (0.315)	0.221 (0.105)	1.053 (0.490)	0.397 (0.369)	0.115 (0.247)	0.188 (0.072)	1.038 (0.311)
RND	0.307 (0.335)	0.157 (0.312)	0.406 (0.174)	1.174 (0.404)	0.350 (0.336)	0.113 (0.248)	0.350 (0.130)	1.197 (0.385)
PPL	0.407 (0.407)	0.177 (0.310)	0.417 (0.164)	0.420 (0.243)	0.480 (0.402)	0.140 (0.249)	0.311 (0.136)	0.353 (0.135)

1. The best performing algorithms are bolded as determined via a 2-Way ANOVA and post-hoc Tukey HSD tests ($p < .05$). No interactions between week and algorithm were found. Full results of statistical testing are available in the repository¹.

2. Max R@10 as $M(SD)$: long courses - 0.351 (0.343), short courses - 0.303 (0.317) • Sample size: long courses - 825, short courses - 268.

mean and standard deviation of measures. The results were averaged over those of each student in each week.

5. RESULTS

Table 2 shows that both CSCLRec and CoPPR achieve high prediction accuracy, while maintaining acceptable diversity and novelty. They outperform their competitors according to precision for both 13-week and 6-week courses. Except for the very similar CoPPR algorithm, CSCLRec's precision is more than 18% higher than that of other recommenders. According to R@10, there is no measurable performance difference among the various algorithms. Considering the maximum possible recall is capped at 35.1% and 30.3%, CSCLRec's R@10 performance (21.9% and 18.8%) suggests it successfully identifies most of the relevant items.

In short courses, two of the best performers according to ILD are the unpersonalized baseline recommenders (RND and PPL) largely due to their introducing randomness. Apart from these random methods, those that emphasize interactions (i.e., MCF and PPR) had better diversity. As a tradeoff to accuracy, CSCLRec's diversity was acceptable - it was somewhere in the middle (3rd of 6 personalized recommenders in ILD) when baseline approaches (RND and PPL) are excluded because they have low precision.

As another tradeoff to high precision, novelty is not best achieved with CoPPR or CSCLRec. Content-based algorithms (i.e., KCB

and SCB) performed well from a novelty perspective as shown through their average MIUF scores. However, they had low diversity scores (ILD); they ranked last or second last.

To illustrate differences in performance over time, we use the LA course as an example (Figure 5). Note similar patterns were present in other courses and the change at week 10 coincides with the term break. In general, CSCLRec and CoPPR remained the best performing recommenders for precision throughout the semester.

Our proposed algorithm, CSCLRec, beats its competitors in precision from weeks 2 or 3 onwards (Figure 5). In contrast, when few inputs from students were available at the beginning of courses, the performance of content-based approaches was worse than the baselines. These results suggest the inclusion of socio-collaborative elements helps address the cold-start problem.

6. DISCUSSION

6.1 Recommender Algorithm Performance

The good performance of content-based recommender algorithms (CB) such as KCB and SCB in recommendation novelty highlights their ability to discover unpopular posts. It implies these approaches are better at helping students more quickly locate difficult-to-find but conceptually related discussions when the goal is to develop narrow but deep knowledge. This class of approaches may also increase forum equity by increasing the visibility of posts made by students with minority opinions that may otherwise go unnoticed in a popularity-based recommender scheme. However, CB algorithms' poor diversity performance suggests they suffer from the over-specialization problem because they only care about content similarity. This makes them unable to recommend semantically diverse resources. Since discussions on the same thread usually have similar content, the suggestions provided by CB recommendation algorithms are likely to direct users to a few specific threads, which may prevent exposure to new ideas. This goes against the general teaching goals of learning contexts where students are expected to discuss and debate different topics.

The family of collaborative filtering algorithms (CF), represented by MCF, showed relatively poor novelty when compared with the CB algorithms. This lack of novelty may discourage the participation of students who hold minority opinions, as has been seen in other investigations [64]. When comparing with PPR, CSCLRec's considerable enhancement in precision demonstrates the effectiveness of its three add-on modules. CoPPR also performs well, but its recommendation diversity appears to be lower. This finding indicates that the design of CSCLRec's content analyzer module benefits recommendation diversity as it is the only

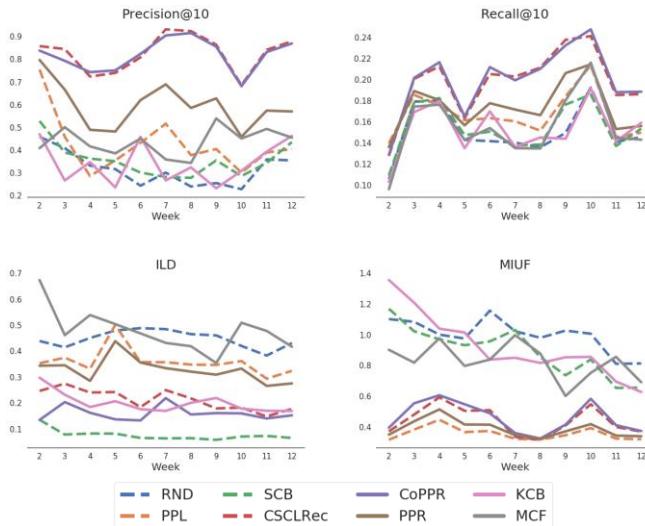


Figure 5. Weekly recommendation results for the LA course.

difference between CSCLRec and CoPPR. The outstanding performance of CSCLRec makes it the most appropriate choice to provide personalized suggestions in small-scale socio-collaborative learning contexts. However, we expect CoPPR to work better than CSCLRec in environments where the domain of discussion topics is narrower than those in our dataset as the use of hypernyms in CSCLRec is prone to mis-classification due to the granularity of the WordNet ontology. For example, in Chemistry, both “Sodium chloride” and “Copper(II) sulfate” are a “chemical compound”, but it makes little sense to link these two terms together as students might be talking about different things.

In contrast to traditional algorithms, CSCLRec and CoPPR integrate pedagogical considerations. The learner interaction profiler module is an obvious case. Unlike the one-size-fits-all recommendation strategy in other algorithms, it employs different strategies (i.e., adding more user-to-user edges) depending on learner type. Compared with the results of other approaches, the user-centered recommendation algorithm design of CSCLRec provided better prediction results by taking advantage of socio-collaborative learning principles.

While CSCLRec tended to perform well in recommendation accuracy, we should acknowledge that such support is not always what is needed for some learner types. For example, listeners not actively engaging in the discussions could be attributed to the recommendation lacking diversity. In this case, collaborative filtering approaches such as MCF might be a better choice. Moreover, new users may benefit from unpersonalized recommenders. For example, PPL could be used when we lack information about that learner because popular discussions may pique newcomer’s interest and encourage them to participate.

6.2 Recommender Support for Learning

The evaluations confirmed that our recommender system can forecast student behavior and give recommendations that match students’ preferences, as represented through their behaviors, in an e-learning discussion forum. Here we discuss the system’s potential to enhance students’ learning processes and outcomes in socio-collaborative learning spaces.

Rooted in learner interest, the generated recommendations can help reduce the time students spend searching for useful resources, thereby increasing the proportion of time dedicated to learning activities (i.e., discussing and sharing). The increased interaction should enable more knowledge-construction within the forum [37, 72], benefiting every learner with more opportunities to review and increase their understanding of the knowledge they have learned [85]. Many empirical studies have also found that student’s active participation in sharing can develop their critical thinking abilities [13] and benefit their overall course performance [19, 61, 84].

At the same time, pedagogical research shows that the diversity and novelty of ideas are critical to learning outcomes, especially during the process of knowledge co-construction. According to the theory of social constructivism, learner exposure to diverse perspectives can help them experience the types of cognitive conflict that lead to knowledge gain [32, 44]. Knowledge building principles also emphasize the importance of diversity and novelty of ideas to the knowledge scaffolding process [69]. Fortunately, CSCLRec’s novelty and diversity performance demonstrated the algorithm’s potential to support various collaborative learning activities in small discussion-based e-learning forums.

6.3 Potential Expansions

There are many ways to further improve the system’s performance when it is deployed online. First, real-time feedback from students can be collected and used to steer the strategies for the next round of recommendations. Second, the system could allow instructor and student configuration. This would allow users to refine the quality of recommendations and offer increased transparency to improve user satisfaction and trust in the recommendation mechanism [82]. In the future, we may adopt a human-in-the-loop approach and let course instructors adjust recommender parameters so they are more consistent with desired teaching plans.

More advanced NLP methods could also be used. For example, using knowledge graphs could benefit graph-based recommenders [56, 57, 59, 63]. Using such approaches could extend the semantic network in the content analyzer. Knowledge graphs relying on Linked Open Data usually have a wider coverage of entities which may allow them to overcome the current algorithms’ lack of phrases for representing key domain-specific concepts [74]. We had tried to use entity linking tools (e.g., DBpedia spotlight [22] and TagMe [31]), to query post content so that key phrases could be linked to entities in the knowledge base which would have replaced the hypernym portion of the PPR graph. However, their performance seemed poor in our context: many key phrases were not linked to the correct knowledge graph entities. The main reason may be that forum posts present disambiguation challenges to entity linking tools [36]. Moreover, some knowledge bases, such as DBpedia [6], have a limited number of verb entities because most verbs are treated as relations. Thus, building a knowledge graph specifically for an individual course seems to be the only realistic approach even though it would require considerable effort.

Lastly, while the proposed recommender performed relatively well, the ability of this recommender to support socio-collaborative learning processes within discussion forums still needs to be validated through in-vivo studies. Due to the limitations of using historical data, the present evaluation does not allow the direct observation of how learners will respond to the recommendations nor does it allow the measurement of the recommendations’ effect on learning processes [28, 30, 51].

7. CONCLUSIONS

In this paper, a novel recommendation approach that accounts for socio-collaborative learning principles in small discussion forums was proposed. This multi-relational graph-based recommendation scheme, CSCLRec, incorporates social network analysis, learner categorization, and natural language processing techniques. A similarly structured recommender, CoPPR, was also introduced for potential use in socio-collaborative learning contexts.

The performance of these proposed algorithms was evaluated in an offline experiment where they were compared against six other recommendation algorithms. The results from this evaluation show our approaches outperform others. Going beyond these measures, we discussed CSCLRec’s potential to help socio-collaborative learning processes, as well as its use cases and potential expansions from the perspective of a variety of measures (e.g., precision, diversity) and learning goals. As future work, we plan to deploy the system to examine its influence on student behaviors and learning.

8. ACKNOWLEDGMENTS

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada [RGPIN-2018-03834].

9. REFERENCES

- [1] Abel, F., Bittencourt, I.I., Costa, E., Henze, N., Krause, D. and Vassileva, J. 2010. Recommendations in online discussion forums for e-learning systems. *IEEE Transactions on Learning Technologies*. 3, 2 (2010), 165–176. DOI:<https://doi.org/10.1109/TLT.2009.40>.
- [2] Abel, F., Bittencourt, I.I., Henze, N., Krause, D. and Vassileva, J. 2008. A Rule-Based Recommender System for Online Discussion Forums. *Adaptive Hypermedia and Adaptive Web-Based Systems* (Berlin, Heidelberg, 2008), 12–21.
- [3] Akyol, Z. and Garrison, D.R. 2008. The development of a community of inquiry over time in an online course: Understanding the progression and integration of social, cognitive and teaching presence. *Journal of Asynchronous Learning Networks*. 12, (2008), 3–22.
- [4] Albatayneh, N.A., Ghauth, K.I. and Chua, F.-F. 2018. Utilizing Learners' Negative Ratings in Semantic Content-based Recommender System for e-Learning Forum. *Journal of Educational Technology & Society*. 21, 1 (2018), 112–125.
- [5] Andresen, M.A. 2009. Asynchronous discussion forums: Success factors, outcomes, assessments, and limitations. *Journal of Educational Technology & Society*. 12, 1 (2009).
- [6] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. *The semantic web*. Springer. 722–735.
- [7] Backenköhler, M., Scherzinger, F., Singla, A. and Wolf, V. 2018. Data-Driven Approach towards a Personalized Curriculum. *Proceedings of the 11th International Conference on Educational Data Mining, EDM 2018* (2018), 246–251.
- [8] Beham, G., Kump, B., Ley, T. and Lindstaedt, S. 2010. Recommending knowledgeable people in a work-integrated learning system. *Procedia Computer Science*. 1, 2 (2010), 2783–2792. DOI:<https://doi.org/10.1016/j.procs.2010.08.003>.
- [9] Bergmeir, C. and Benítez, J.M. 2012. On the use of cross-validation for time series predictor evaluation. *Information Sciences*. 191, (2012), 192–213. DOI:<https://doi.org/10.1016/j.ins.2011.12.028>.
- [10] Bobadilla, J., Ortega, F., Hernando, A. and Bernal, J. 2012. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-based systems*. 26, (2012), 225–238.
- [11] Breese, J.S., Heckerman, D. and Kadie, C. 1998. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (San Francisco, CA, USA, 1998), 43–52.
- [12] Buder, J. and Schwind, C. 2012. Learning with personalized recommender systems: A psychological view. *Computers in Human Behavior*. 28, 1 (Jan. 2012), 207–216. DOI:<https://doi.org/10.1016/j.chb.2011.09.002>.
- [13] Carini, R.M., Kuh, G.D. and Klein, S.P. 2006. Student engagement and student learning: Testing the linkages. *Research in Higher Education*. 47, 1 (Feb. 2006), 1–32. DOI:<https://doi.org/10.1007/s11162-005-8150-9>.
- [14] Castells, P., Hurley, N.J. and Vargas, S. 2015. Novelty and diversity in recommender systems. *Recommender systems handbook*. Springer. 881–918.
- [15] Catherine, R. and Cohen, W. 2016. Personalized Recommendations Using Knowledge Graphs: A Probabilistic Logic Programming Approach. *Proceedings of the 10th ACM Conference on Recommender Systems* (New York, NY, USA, 2016), 325–332.
- [16] Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C. and others 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*. (2018).
- [17] Chen, B., Chang, Y.H., Ouyang, F. and Zhou, W. 2018. Fostering student engagement in online discussion through social learning analytics. *Internet and Higher Education*. 37, (2018), 21–30. DOI:<https://doi.org/10.1016/j.iheduc.2017.12.002>.
- [18] Chen, W. and Persen, R. 2009. A recommender system for collaborative knowledge. *Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling* (2009), 309–316.
- [19] Cheng, C.K., Paré, D.E., Collimore, L.-M. and Joordens, S. 2011. Assessing the effectiveness of a voluntary online discussion forum on improving students' course performance. *Computers & Education*. 56, 1 (2011), 253–261. DOI:<https://doi.org/https://doi.org/10.1016/j.compedu.2010.07.024>.
- [20] Coxhead, A. 2000. A New Academic Word List. *TESOL Quarterly*. 34, 2 (2000), 213. DOI:<https://doi.org/10.2307/3587951>.
- [21] Cremonesi, P., Koren, Y. and Turrin, R. 2010. Performance of recommender algorithms on top-N recommendation tasks. *RecSys '10 - Proceedings of the 4th ACM Conference on Recommender Systems* (New York, New York, USA, 2010), 39–46.
- [22] Daiber, J., Jakob, M., Hokamp, C. and Mendes, P.N. 2013. Improving Efficiency and Accuracy in Multilingual Entity Extraction. *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)* (2013).
- [23] Demmans Epp, C., Phirangee, K. and Hewitt, J. 2017. Student actions and community in online courses: The roles played by course length and facilitation method. *Online Learning Journal*. 21, 4 (2017).
- [24] Demmans Epp, C., Phirangee, K. and Hewitt, J. 2017. Talk with Me: Student Behaviours and Pronoun Use as Indicators of Discourse Health across Facilitation Methods. *Journal of Learning Analytics*. 4, 3 (Dec. 2017), 47–75. DOI:<https://doi.org/10.18608/jla.2017.43.4>.
- [25] Dowell, N.M.M., Brooks, C., Kovanović, V., Joksimović, S. and Gašević, D. 2017. The changing patterns of MOOC discourse. *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale* (2017), 283–286.
- [26] Dowell, N.M.M., Skrypnyk, S., Joksimović, S., Graesser, A., Dawson, S., Gašević, D., Hennis, T.A., Vries, P. de

- and Kovanović, V. 2015. Modeling Learners' Social Centrality and Performance through Language and Discourse. *Proceedings of the 8th International Conference on Educational Data Mining, EDM 2015* (2015), 250–257.
- [27] Drachsler, H., Verbert, K., Santos, O.C. and Manouselis, N. 2015. Panorama of Recommender Systems to Support Learning. *Recommender Systems Handbook*. F. Ricci, L. Rokach, and B. Shapira, eds. Springer US. 421–451.
- [28] Erdt, M., Fernández, A. and Rensing, C. 2015. Evaluating Recommender Systems for Technology Enhanced Learning: A Quantitative Survey. *IEEE Transactions on Learning Technologies*. 8, 4 (2015), 326–344. DOI:https://doi.org/10.1109/TLT.2015.2438867.
- [29] Esteban, A., Zafra, A. and Romero, C. 2018. A Hybrid Multi-Criteria Approach Using a Genetic Algorithm for Recommending Courses to University Students. *Proceedings of the 11th International Conference on Educational Data Mining, EDM 2018* (2018), 273–279.
- [30] Fazeli, S., Drachsler, H., Bitter-Rijkema, M., Brouns, F., Van Der Vegt, W. and Sloep, P.B. 2018. User-Centric Evaluation of Recommender Systems in Social Learning Platforms: Accuracy is Just the Tip of the Iceberg. *IEEE Transactions on Learning Technologies*. 11, 3 (2018), 294–306. DOI:https://doi.org/10.1109/TLT.2017.2732349.
- [31] Ferragina, P. and Scaiella, U. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). *Proceedings of the 19th ACM international conference on Information and knowledge management* (2010), 1625–1628.
- [32] Fosnot, C.T. and Perry, R.S. 1996. Constructivism: A psychological theory of learning. *Constructivism: Theory, perspectives, and practice*. 2, (1996), 8–33.
- [33] García, E., Romero, C., Ventura, S. and Castro, C. De 2009. An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. *User Modeling and User-Adapted Interaction*. 19, 1-2 SPEC. ISS. (2009), 99–132. DOI:https://doi.org/10.1007/s11257-008-9047-z.
- [34] Greer, J., McCalla, G., Cooke, J., Collins, J., Kumar, V., Bishop, A. and Vassileva, J. 1998. The Intelligent Helpdesk: Supporting Peer-Help in a University Course. *International Conference on Intelligent Tutoring Systems*. Springer Verlag. 494–503.
- [35] Haveliwala, T.H. 2002. Topic-sensitive PageRank. *Proceedings of the 11th International Conference on World Wide Web, WWW '02* (New York, New York, USA, 2002), 517–526.
- [36] Heitmann, B. and Hayes, C. 2010. Using linked data to build open, collaborative recommender systems. *AAAI Spring Symposium - Technical Report*. SS-10-07, October 2010 (2010), 76–81.
- [37] Hew, K.F. and Cheung, W.S. 2012. *Student participation in online discussions: Challenges, solutions, and future research*. Springer Science & Business Media.
- [38] Hewitt, J. 2005. Toward an understanding of how threads die in asynchronous computer conferences. *Journal of the Learning Sciences*. 14, 4 (2005), 567–589. DOI:https://doi.org/10.1207/s15327809jls1404_4.
- [39] Hmelo-Silver, C. 2013. *The International Handbook of Collaborative Learning*. Routledge.
- [40] Hu, Y., Koren, Y. and Volinsky, C. 2008. Collaborative Filtering for Implicit Feedback Datasets. *2008 Eighth IEEE International Conference on Data Mining* (Dec. 2008), 263–272.
- [41] Ishola, O.M. and McCalla, G. 2017. Predicting prospective peer helpers to provide just-in-time help to users in question and answer forums. *Proceedings of the 10th International Conference on Educational Data Mining, EDM 2017* (2017), 238–243.
- [42] Jeong, H. and Hmelo-Silver, C.E. 2016. Seven Affordances of Computer-Supported Collaborative Learning: How to Support Collaborative Learning? How Can Technologies Help? *Educational Psychologist*. 51, 2 (Apr. 2016), 247–265. DOI:https://doi.org/10.1080/00461520.2016.1158654.
- [43] Jones, D., Bench-Capon, T. and Visser, P. 1998. Methodologies for Ontology Development. *Proc. IT&KNOWS Conference of the 15th IFIP World Computer Congress*. (1998).
- [44] Kamii, C. 1986. The Equilibration of Cognitive Structures: The Central Problem of Intellectual Development. Jean Piaget, Terrance Brown, Kishore Julian Thampy. *American Journal of Education*. 94, 4 (Aug. 1986), 574–577. DOI:https://doi.org/10.1086/443876.
- [45] Kleanthous Loizou, S. and Dimitrova, V. 2013. Adaptive notifications to support knowledge sharing in close-knit virtual communities. *User Modeling and User-Adapted Interaction*. 23, 2–3 (2013), 287–343. DOI:https://doi.org/10.1007/s11257-012-9127-y.
- [46] Kleanthous, S. and Dimitrova, V. 2010. Analyzing community knowledge sharing behavior. *International Conference on User Modeling, Adaptation, and Personalization* (2010), 231–242.
- [47] Kleanthous, S. and Dimitrova, V. 2009. Detecting changes over time in a knowledge sharing community. *Proceedings - 2009 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2009*. 1, (2009), 100–107. DOI:https://doi.org/10.1109/WI-IAT.2009.21.
- [48] Kleanthous, S. and Dimitrova, V. 2008. Modelling Semantic Relationships and Centrality to Facilitate Community Knowledge Sharing. *Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer Berlin Heidelberg. 123–132.
- [49] Lan, A.S., Spencer, J.C., Chen, Z., Brinton, C.G. and Chiang, M. 2019. Personalized Thread Recommendation for MOOC Discussion Forums. *Machine Learning and Knowledge Discovery in Databases* (Cham, 2019), 725–740.
- [50] Lesk, M. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. *Proceedings of the 5th Annual International Conference on Systems Documentation, SIGDOC 1986* (New York, New York, USA, Jun. 1986), 24–26.

- [51] Manouselis, N., Drachsler, H., Verbert, K. and Duval, E. 2012. *Recommender Systems for Learning*. Springer Publishing Company, Incorporated.
- [52] McMillan, C., Poshyvanyk, D. and Grechanik, M. 2010. Recommending source code examples via api call usages and documentation. *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering* (2010), 21–25.
- [53] Mi, F. and Faltings, B. 2017. Adaptive Sequential Recommendation for Discussion Forums on MOOCs using Context Trees. *Proceedings of the 10th International Conference on Educational Data Mining*. (2017), 24–31.
- [54] Mihalcea, R. and Tarau, P. 2004. TextRank: Bringing Order into Text. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* (Barcelona, Spain, 2004), 404–411.
- [55] Miller, G.A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*. 38, 11 (Nov. 1995), 39–41. DOI:https://doi.org/10.1145/219717.219748.
- [56] Musto, C., Basile, P., Lops, P., de Gemmis, M. and Semeraro, G. 2017. Introducing linked open data in graph-based recommender systems. *Information Processing & Management*. 53, 2 (2017), 405–435. DOI:https://doi.org/https://doi.org/10.1016/j.ipm.2016.12.003.
- [57] Musto, C., Lops, P., Basile, P., de Gemmis, M. and Semeraro, G. 2016. Semantics-Aware Graph-Based Recommender Systems Exploiting Linked Open Data. *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization* (New York, NY, USA, 2016), 229–237.
- [58] Musto, C., Semeraro, G., de Gemmis, M. and Lops, P. 2017. Tuning Personalized PageRank for Semantics-Aware Recommendations Based on Linked Open Data. *The Semantic Web* (Cham, 2017), 169–183.
- [59] Nguyen, P.T., Tomeo, P., Di Noia, T. and Di Sciascio, E. 2015. An evaluation of simrank and personalized pagerank to build a recommender system for the web of data. *WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web* (2015), 1477–1482.
- [60] Page, L., Brin, S., Motwani, R. and Winograd, T. 1998. The PageRank Citation Ranking: Bringing Order to the Web. *World Wide Web Internet And Web Information Systems*. 54, 1999–66 (1998), 1–17. DOI:https://doi.org/10.1.1.31.1768.
- [61] Palmer, S., Holt, D. and Bray, S. 2008. Does the discussion help? the impact of a formally assessed online discussion on final student results. *British Journal of Educational Technology*. 39, 5 (Sep. 2008), 847–858. DOI:https://doi.org/10.1111/j.1467-8535.2007.00780.x.
- [62] Park, Y.J. and Tuzhilin, A. 2008. The long tail of recommender systems and how to leverage it. *RecSys'08: Proceedings of the 2008 ACM Conference on Recommender Systems* (New York, New York, USA, 2008), 11–18.
- [63] Pereira, C.K., Campos, F., Ströele, V., David, J.M.N. and Braga, R. 2018. BROAD-RSI – educational recommender system using social networks interactions and linked data. *Journal of Internet Services and Applications*. 9, 1 (2018), 7. DOI:https://doi.org/10.1186/s13174-018-0076-5.
- [64] Phirangee, K., Demmans Epp, C. and Hewitt, J. 2016. Exploring the Relationships between Facilitation Methods, Students' Sense of Community, and Their Online Behaviors. *Online Learning*. 20, 2 (2016), 134–154.
- [65] Polyzou, A., Athanasios, N. and Karypis, G. 2019. Scholars Walk: A Markov Chain Framework for Course Recommendation. *Proceedings of the 12th International Conference on Educational Data Mining* (2019), 396–401.
- [66] Rosé, C.P. and Ferschke, O. 2016. Technology Support for Discussion Based Learning: From Computer Supported Collaborative Learning to the Future of Massive Open Online Courses. *International Journal of Artificial Intelligence in Education*. 26, 2 (Jun. 2016), 660–678. DOI:https://doi.org/10.1007/s40593-016-0107-y.
- [67] Rose, S., Engel, D., Cramer, N. and Cowley, W. 2010. Automatic Keyword Extraction from Individual Documents. *Text Mining: Applications and Theory*. John Wiley and Sons. 1–20.
- [68] Santos, O.C., Boticario, J.G. and Pérez-Marín, D. 2014. Extending web-based educational systems with personalised support through User Centred Designed recommendations along the e-learning life cycle. *Science of Computer Programming*. 88, (Aug. 2014), 92–109. DOI:https://doi.org/10.1016/j.scico.2013.12.004.
- [69] Scardamalia, M. 2002. Collective cognitive responsibility for the advancement of knowledge. *Liberal education in a knowledge society*. (2002), 67–98.
- [70] Scardamalia, M. and Bereiter, C. 1994. Computer Support for Knowledge-Building Communities. *Journal of the Learning Sciences*. 3, 3 (Jul. 1994), 265–283. DOI:https://doi.org/10.1207/s15327809jls0303_3.
- [71] Scardamalia, M. and Bereiter, C. 2006. Knowledge Building: Theory, Pedagogy, and Technology. *The Cambridge handbook of: The learning sciences*. Cambridge University Press. 97–115.
- [72] Schellens, T. and Valcke, M. 2006. Fostering knowledge construction in university students through asynchronous discussion groups. *Computers and Education*. 46, 4 (2006), 349–370. DOI:https://doi.org/10.1016/j.compedu.2004.07.010.
- [73] Shani, G. and Gunawardana, A. 2011. Evaluating Recommendation Systems. *Recommender Systems Handbook*. Springer US. 257–297.
- [74] Shen, W., Wang, J. and Han, J. 2015. Entity Linking with a Knowledge Base : *IEEE Transactions on Knowledge and Data Engineering*. 27, 2 (2015), 443–460. DOI:https://doi.org/10.1109/TKDE.2014.2327028.
- [75] Smith, L. and Macgregor, J.T. 1992. What is Collaborative Learning ? *Assessment*. 117, 5 (1992), 10–30.

- [76] Smyth, B. and McClave, P. 2001. Similarity vs. diversity. *International conference on case-based reasoning* (2001), 347–361.
- [77] Steck, H. 2011. Item popularity and recommendation accuracy. *RecSys'11 - Proceedings of the 5th ACM Conference on Recommender Systems* (New York, New York, USA, 2011), 125–132.
- [78] Thai-Nghe, N., Drumond, L., Horváth, T., Krohn-Grimberghe, A., Nanopoulos, A. and Schmidt-Thieme, L. 2012. Factorization techniques for predicting student performance. *Educational recommender systems and technologies: Practices and challenges*. IGI Global. 129–153.
- [79] Vassileva, J. 2008. Toward Social Learning Environments. *IEEE Transactions on Learning Technologies*. 1, 4 (2008), 199–214. DOI:<https://doi.org/10.1109/TLT.2009.4>.
- [80] Vassileva, J., McCalla, G.I. and Greer, J.E. 2016. From Small Seeds Grow Fruitful Trees: How the PHelpS Peer Help System Stimulated a Diverse and Innovative Research Agenda over 15 Years. *International Journal of Artificial Intelligence in Education*. 26, 1 (2016), 431–447. DOI:<https://doi.org/10.1007/s40593-015-0073-9>.
- [81] Veldhuis-Diermanse, A.E. 2002. *Participation, learning activities and knowledge construction in computer-supported collaborative learning in higher education*.
- [82] Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I. and Duval, E. 2012. Context-aware recommender systems for learning: A survey and future challenges. *IEEE Transactions on Learning Technologies*.
- [83] Vygotsky, L.S. 1978. *Mind in society: Development of higher psychological processes*. Harvard University Press.
- [84] Webb, E., Jones, A., Barker, P. and Van Schaik, P. 2004. Using e-learning dialogues in higher education. *Innovations in Education and Teaching International*. 41, 1 (2004). DOI:<https://doi.org/10.1080/1470329032000172748>.
- [85] Weber, K., Maher, C., Powell, A. and Lee, H.S. 2008. Learning opportunities from group discussions: Warrants become the objects of debate. *Educational Studies in Mathematics*. 68, 3 (2008), 247–261.
- [86] Wise, A.F., Hausknecht, S.N. and Zhao, Y. 2014. Attending to others' posts in asynchronous discussions: Learners' online "listening" and its relationship to speaking. *International Journal of Computer-Supported Collaborative Learning*. 9, 2 (May 2014), 185–209. DOI:<https://doi.org/10.1007/s11412-014-9192-9>.
- [87] Yang, D., Piergallini, M., Howley, I. and Rose, C.P. 2014. Forum Thread Recommendation for Massive Open Online Courses. *Proceedings of the 7th International Conference on Educational Data Mining*. (2014), 257–260. DOI:https://doi.org/https://www.cs.cmu.edu/~diyiy/docs/edm14_recom.pdf.
- [88] Zhang, S., Yao, L., Sun, A. and Tay, Y. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 52, 1 (2019). DOI:<https://doi.org/10.1145/3285029>.

Deep Embeddings of Contextual Assessment Data for Improving Performance Prediction

Benjamin Clavié

School of Informatics, University of Edinburgh
Edinburgh, Scotland
benjamin.clavie@ed.ac.uk

Kobi Gal

School of Informatics, University of Edinburgh
Edinburgh, Scotland
kgal@ed.ac.uk

ABSTRACT

We introduce DeepPerfEmb, or DPE, a new deep-learning model that captures dense representations of students' on-line behaviour and meta-data about students and educational content. The model uses these representations to predict student performance. We evaluate DPE on standard datasets from the literature, showing superior performance to the state-of-the-art systems in predicting whether or not students will answer a given question correctly. In particular, DPE is unaffected by the cold-start problem which arises when new students come to the system with little to no data available. We also show strong performance of the model when removing students' histories altogether, relying in part on contextual information about the questions. This strong performance without any information about the learners' histories demonstrates the high potential of using deep embedded representations of contextual information in educational data mining.

1. INTRODUCTION

The *testing* effect, the effect of including practice assessments as part of a students' learning phase, is known to have a strong positive influence on the knowledge acquisition process [2].

While the importance of regular practice and question answering is established, it is essential to balance it against the time constraints that students and instructors are facing [11]. The issue of having to teach and evaluate "*too much [...] in too short a time*" [10] is long-standing and leads to teachers having to make instructional choices with the information they have available [12]. It is thus important to identify factors that could help intelligent systems to ask the right question to the right students to maximise their knowledge gain in a limited time.

Extensive research has focused on building better student modeling to work towards this goal. Most of these approaches focus on extracting information from individuals'

histories of answers given, both right and wrong, to questions evaluating certain skills [4, 19, 18, 7]. Recent work has taken into account other factors, such as item-skills relationships, the relationship between a question and the skill it is meant to evaluate citedas3h, or individual item difficulty [17] in predicting student performance.

Deep knowledge tracing, which represents the state of the art in student performance, does not take into account the wealth of instance-specific interactions a student has with a given question, such as requesting assistance before attempting to answer it or the amount of time taken before answering.

We propose DeepPerfEmb, a deep learning model whose aim is to learn dense representations of this information and use it to improve the task of performance prediction. Our contribution is two-fold: we firstly argue that instance-specific information can be leveraged by such a model to reach a very high level of performance on predicting student correctness. We also introduce a variant of the model using exclusively contextual data, showing its ability to learn dense representations of these data points and perform strongly on the same task, despite having very limited information about the students' actions.

2. BACKGROUND

In the educational data mining field, there has been extensive research on attempting to model a learner's understanding of defined skills. Generally, this task is achieved through using observations related to a student's question-answering history. This information is used to estimate the student's mastery of the skills evaluated by the questions and is generally evaluated by using the model to predict whether or not they will answer a given question correctly. Such models are known as *Knowledge Tracing* (KT) models. *Bayesian Knowledge Tracing* (BKT), one of the most widespread classical method, models each students' knowledge as the latent variable of a Hidden Markov Model built using students' answering histories [4]. Such methods also rely on an evaluation of the probability of *slipping*, when a student answers incorrectly despite having mastered the skill, and *guessing*, when a correct answer is given without having mastered it.

More recently, many different approaches to knowledge tracing have been researched, mainly relying on extracting information from a vast amount of students' attempts at answering questions [7, 18]. Some of these models occasionally

Benjamin Clavié and Kobi Gal "Deep Embeddings of Contextual Assessment Data for Improving Performance Prediction" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 374 - 380

focus on or integrate other factors, such as modelling student forgetting [23] and estimated difficulty of question [15] or the possibility for a single question to relate to multiple, distinct types of knowledge [3]. These approaches often serve as the basis to intelligent learning schedulers, aiming to optimise the distribution of questions asked to students to maximise their knowledge gain [22, 25].

In recent years, deep learning has been utilised in order to produce better-performing variants of previous approaches. Notably, DeepIRT [27] and Deep Knowledge Tracing [19], have been introduced. These techniques, themselves a refinement on previous models, replace some of the prior building blocks with deep neural architectures while retaining the same foundational approach. Unlike more traditional methods, deep-learning based approaches rarely explicitly model the impact of forgetting, guessing or slipping, instead relying on the model to capture implicit information about these factors.

Online intelligent tutoring systems, such as the Assistment platform [20], have been invaluable in providing a large amount of data to train and evaluate such models. In addition to the information about students' attempts, failures and successes in answering questions, they generate a wealth of data about other aspects of the tutoring system. Notably, such systems may provide the user with the possibility of requesting assistance in answering the question, in the form of hints. It has been noted that such additional features are under-utilised in KT models and improve their performance when taken into account [26].

The focus of most of this prior work has been on exploiting the history of user answers, both right and wrong, in order to predict the likelihood that they have mastered a given skill. Such approaches reach a high level of performance and can accurately model the relationships between the skills evaluated [19, 16]. However, they encounter issues with students with relatively little or no interaction, and some of them exclude any student who has attempted to answer fewer than 10 questions [15, 3]. This issue is known as the *cold start problem*.

However, point-of-time snapshots of data contain a lot of additional information that has known little exploration. Such information, which we broadly refer to as **contextual information**, includes data directly related to the students' context, such as their school, the question they are solving, and the time it takes them to attempt to answer a question. We believe that such a method is complementary to approaches focusing on students' history in understanding the cognitive process of learning through assessment.

Prior work on deep neural networks has highlighted their ability to learn good embedding representations for discrete data [6]. This paper demonstrates that a modified version of this approach is able to outperform state-of-the-art KT model in the specific task of predicting student correctness. We show that our model learns a powerful representation of the data it receives as input, outperforming the state of the art, leading to a better understanding of how the questions asked to students can affect their performance.

3. PROPOSED METHOD

Our goal is to highlight how contextual data can be leveraged to improve question-correctness prediction. In order to do so, we use a deep learning model whose main purpose is to learn representations of this data in order to predict question-correctness. We then set out to leverage interpretation methods in order to understand which factors are considered important in making these predictions.

3.1 Data

We use two widely used public datasets made available by the Assistments online tutoring platform [20]: **ASSIST2009** [5] and **ASSISTChall** [1].

Each dataset is composed of hundreds of thousands of student interaction, with each interaction corresponding to a snapshot taken at the moment a student attempts to answer a question. Each snapshot contains a large amount of information, represented by multiple variables.

Two categories of data are present in each snapshot:

- **Meta-data, or contextual data:** Information about the overall context around the student and the question they are currently taking. Broadly, these are:
 - Information about the student's background (school ID, teacher ID...)
 - Information about the current question (problem set ID, question ID, skill evaluated ID, whether or not the question can be *scaffolded*...)
- **Current instance-specific data:** Information about the question the student is currently attempting. Broadly, these are:
 - Information about the student's help requests (hints requested, whether he has seen the final hint, where the questions stands in a *scaffolding*...)
 - Information about the time spent on the current question (time before first interaction, total time with question...)

Both datasets do not contain exactly the same information. ASSIST09 contains additional information in the form of both **interaction data**, such as time-to-first action and total time on question, and **contextual meta data**, notably relative to individual students' background, such as the specific assignment set they are working on or the ID of their class. Additionally, ASSISTChall is notable due to the presence of **scaffolded** questions. Scaffolding is an alternative to hints in making it simpler for a student to answer a harder question [21]. A scaffolded question is a question that can be decomposed into simpler questions (the *scaffolding questions*). The data contains variables describing the scaffolding status of an interaction: whether a question is the start of a scaffolding and whether it is part of one. For the purpose of our experiments, we consider scaffolding to be a type of contextual data as an attribute of the question being asked.

Due to the nature of the information contained in our snapshots, they contain both **categorical** and **continuous** variables:

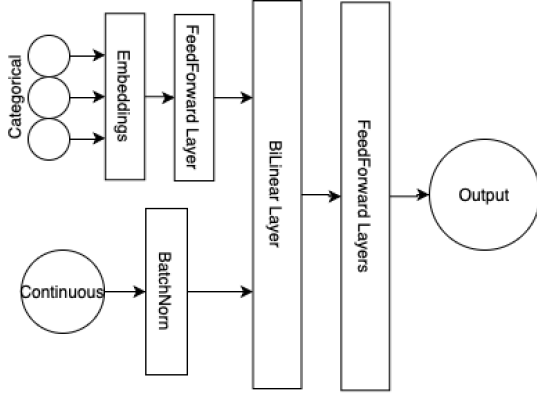


Figure 1: Simplified view of full model

Categorical variables, in this case, represent information that belongs to a finite number of defined categories, such as the skill being evaluated, the ID of the problem set the student is working through or the first action that they took on the current question (whether he requested a hint or attempted to answer it).

Continuous variables, on the other hand, represent information that can be measured, such as how long it takes for the user to first interact with the question after seeing it. For this work, ordinal variables, such as how many hints a student has received, are treated the same as continuous variables.

3.2 Preprocessing

We apply four major preprocessing steps to the data. For all of them except the removal of non-attempt snapshots, we use the data preprocessing utilities in the fastai2 library [8].

3.2.1 Removal of information leaks

Both datasets contain some variables that are perfectly correlated with student correctness. These are values such as the **hint** variable, which indicates that this interaction resulted in the user requesting a hint instead of trying to answer the question. The system will automatically label this interaction as "incorrect", although no attempt was made. As we do not want the model to learn incorrect information from this data and reach an artificially high score, these interactions are removed from the data.

Additionally, we also remove the variables that could lead to our model learning about an individual's student history. This includes the user ID, the total count of attempts by a user, the exact timestamp of interactions as well as additional information contained in ASSISTChall, such as a student's career path, final test score or emotional state.

3.2.2 Standardisation of Continuous Data

All the continuous variables are normalised before being fed to the model.

3.2.3 Handling Missing Continuous Value

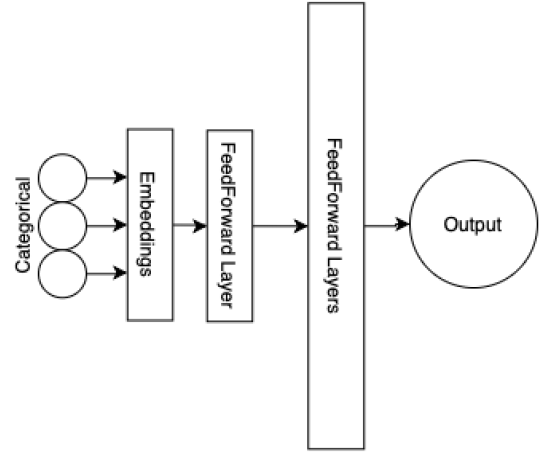


Figure 2: Simplified view of meta-data only model

In some cases, all continuous variables are not available in a given snapshot. In order to account for this factor, we create a categorical variable corresponding to each continuous variable. This variable represents whether the information is present in the current snapshot or not. This allows the model to potentially capture the meaning of the absence of a given observation in a snapshot.

3.2.4 Pre-encoding of Categorical Data

Prior to being passed as input to the model, all categorical variables are ordinally encoded. This means that each possible value is replaced by an integer representing it. This step is crucial in ensuring the model can learn a dense representation of each possible value during training.

3.3 Model

Predicting the performance of a student based on a student's previous answers on questions meant to evaluate defined skills has been widely explored in work on *Knowledge Tracing*. Our aim is to build a model learning good representations of data without individual students' histories to predict whether or not a student will answer a question correctly.

Our model is a variant of the model presented in [6] with several modifications. The overall architecture can be described as follows.

3.3.1 Architecture

Structure

Embeddings: We create an embedding layer for each of the categorical variables we are processing. This embedding process uses a function e_i , which maps each possible categorical input x_i to a corresponding dense vector X_i :

$$e_i : x_i \mapsto X_i \quad (1)$$

This means that each of the categorical variables C will be mapped to a vector space. Each embedding is learned during the model training, and our aim is for the model to learn a representation of the categorical variables describing a given snapshot.

This step is the key step of our network, as the embeddings are trained alongside the full network during model training. With the task of predicting student correctness as its final objective, the model will use these embeddings layer to learn a representation for each of the variables it is given as an input.

Finally, the embedded representation of all the categorical variables are concatenated together into a single vector. This vector is then passed to a single feedforward layer, as defined below.

Bilinear Layer: The authors of [6] concatenated the normalised continuous inputs with the previously generated concatenation of the categorical variables. This approach resulted in unstable training and overfitting on ASSIST09. To alleviate this and allow our model to better weigh both types of features, we introduce a Bilinear layer.

The Bilinear layer takes two vectors as input, x and y , and turns them into a single output vector by multiplying them with a learned weight w and adding a learned bias b . The activation function and batch normalisation functions are both applied to this and every subsequent layers:

$$\text{BatchNorm}(\text{Mish}(x * w * y + b)) \quad (2)$$

FeedForward layers: The inputs are then passed through a classical feedforward architecture made up of linear layers which multiply the single input vector x by a learned weight w and add a learned bias b :

$$\text{BatchNorm}(\text{Mish}(x * w + b)) \quad (3)$$

Output layer: Our output layer is a normal feedforward layer with two output nodes, representing the prediction made by the model (correct or incorrect).

For the experiments exploiting both interaction and meta-data, we use the full version of our model as presented in Figure 1. When using only the meta-data, which is expressed through categorical variables exclusively, we do not need the weighing introduced by the Bilinear layer to allow the model to converge. As a result, in this situation, we use a simplified architecture presented in Figure 2.

Information

Activation: Our model uses the Mish activation function, which has been shown to consistently outperform common activation functions such as ReLU [14].

Batch Normalisation: It has previously been demonstrated that batch normalisation helps in both stabilising and speeding up the training of neural networks [9]. As such, we apply batch normalisation to our continuous input and to the output of every other layer.

Dropout: To prevent overfitting, which happens when the model learns too much about the training data and fails to generalise, dropout [24] is applied after every layer. We applied a dropout value of 0.4 during our experiments.

4. EXPERIMENTAL SETTING

We separate our experimentation into two parts. Firstly, we will use both of the data types we defined earlier, **meta-data** and **instance specific data**. This experiment will serve as a first indicator of our model’s ability to extract

information from the data and build efficient representation. We will then perform feature importance analysis on the models’ predictions to understand what variables have the strongest impact on its predictions.

Following this, we will attempt to predict question-correctness using exclusively **meta-data**. The aim of this experiment is to highlight how much the model can learn while using no information about the current assessment session or the learner’s history. We will then study the model to understand what representation of the data it has learned and how it impacts its performance.

We evaluate our model by performing 5-fold cross-validation and training the model for 100 epochs on each of the steps, saving and reporting the result obtained for the best epoch. For both datasets, we use the LAMB optimiser [29], which is better suited to large-batches training than other optimisers. In order to minimise training time, batch size is set to 24 000 and a maximum learning rate of 10^{-1} is used. In both models, we set the hidden dimensions of all layers to 100. These hyperparameters were obtained by a search using the first fold of the cross-validation set.

Due to the imbalanced distribution of our data, we report prediction results using the Area Under the receiver-operator Curve (**AUC**) metric, widely used in the literature for similar tasks [19, 3, 28].

For reference purposes, we have included results from the two most widespread implementations of Knowledge Tracing, BKT and DKT (here, DKT+ [28], a slight refinement of standard DKT) as well as from the current state-of-the-art, SAKT [16] in the comparison tables. For BKT, we use the best results reported in the paper introducing DKT [19]. Although the original data used by DPE and KT models is the same, we use different information found in the datasets. KT models use individual students’ interaction histories in order to predict performance and discard the rest of the information. On the other hand, DPE focuses on the contextual data and explicitly avoids the use of any student history data. As such, the scores are given in order to compare their results when focusing exclusively on the task of predicting question-correctness, but are not directly comparable as KT models leverage this task as a way to model student behaviour whereas our aim is to evaluate the importance of other, individual-unrelated features.

4.1 Using Instance Specific and Meta-Data

We first attempt to build a performance predictor using the two types of data we defined earlier, **contextual meta-data** and **instance specific data**. This model is likely to perform well, as it has access to a vast array of information about the current question as well as instance information such as the amount and type of help requested, the time before an action is taken as well as the total time spent on the current question.

4.2 Using Meta-Data

Our second experiment focuses on using exclusively the data we defined earlier as **meta-data**. This means that we remove interaction data from the input data.

Table 1: Results

Model	All-data		Meta-data	
	ASSIST2009 AUC	ASSISTChall AUC	ASSIST2009 AUC	ASSISTChall AUC
DPE (Ours)	0.87	0.76	0.75	0.63
BKT (reference)	0.69	N/A	0.69	N/A
DKT+ (reference)	0.82	0.73	0.82	0.73
SAKT (reference)	0.84	0.73	0.84	0.73

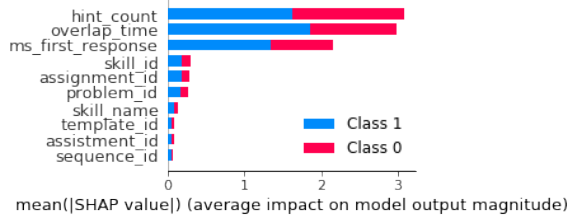


Figure 3: SHAP Values for ASSIST09 (all data)

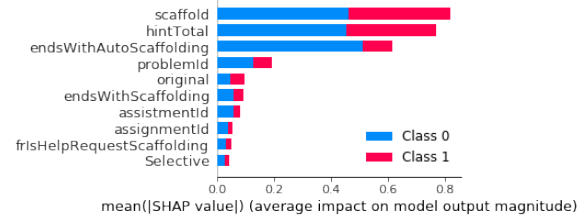


Figure 4: SHAP Values for ASSISTChall (all data)

We do so in order to force the classifier to learn strong representations of contextual meta-data about the student and the question themselves. Reaching a good level of performance using such limited data would suggest that these representations could be exploited to discover new insights about assessment and be combined with traditional knowledge tracing techniques to develop better assignments.

4.3 Interpreting Results And Feature Importances

Following the evaluation of the classifiers, we will attempt to extract information about the factors that strongly influence our model.

We will interpret the model’s predictions using Deep Shapley Additive Explanations (DeepSHAP) [13]. By randomly replacing the values of subsets of the input features by uninformative values, DeepSHAP measures the influence of each input feature on different parts of a deep neural network and produces SHAP values for each prediction examples. SHAP values are an estimation of the importance of the feature in the prediction of each label made by the model.

We run DeepSHAP on randomly selected representative examples from the validation set and report the mean SHAP values of the features over all the examples, providing a visualisation of the features used by the model in its prediction. In all figures, class 0, the negative class, refers to a student answering a question incorrectly while class 1 refers to them having successfully answered the question. Although deep learning models remain black boxes and such interpretation techniques are vulnerable to adversarial examples, they provide a solid base towards making sense of model predictions.

5. RESULTS AND DISCUSSION

The results for this experiment are presented in Table 1, with BKT, DKT+ and SAKT results also presented for reference purposes.

When using all the available data, our approach performs extremely well in predicting question-correctness on ASSIST2009, reaching an AUC of 0.87 on ASSIST2009 and 0.76 on ASSISTChall, slightly outperforming state-of-the-art KT ap-

proaches for this task.

Our approach also reaches relatively high AUCs scores of 0.75 and 0.63 on ASSIST09 and ASSISTChall, respectively, when removing the instance-specific interaction data and using meta-data exclusively. This suggests that the models, while not outperforming student history-based methods, are able to extract enough information from contextual meta-data to reach a good level of performance, even outperforming the reported BKT results for ASSIST2009. In order to better understand what factors drive the models’ performance, we will compute the SHAP value corresponding to an estimate of the importance of each feature.

The SHAP values for the models exploiting the full data are presented in Figure 3 and 4. In ASSIST09, the temporal features, detailing how long the student has been interacting with the current question and how long until they first interact with the question, are of high importance.

More notably, on both datasets, the features that appear to be the most influential focus measuring the amount of help a student has needed to answer the current question. Features related to **hints**, such as the amount of hints requested for the current question (*hint_count* and *hint_total*), have a very strong influence on predictions. As hints are automatically given in case of failure, the hint-related features also capture information about the number of attempts made on the current question during the current question.

In ASSISTChall, features related to **scaffolding**, another form of assistance the student can receive, also have strong influence on the prediction, further supporting the importance of assistance factors.

The figure also shows that the other variables which we described as **meta-data**, such as the problem ID, do play a role predicting question-correctness, with a stronger impact on the likelihood of a question being answered incorrectly than correctly. We explore the influence of these factors further in Figure 5 and 6, showing SHAP values for the models which only use contextual meta-data.

In the case of ASSISTment, we notice that problems with the ability to end in auto-scaffolding are a strong predictor on whether or not a student will correctly answer a question.

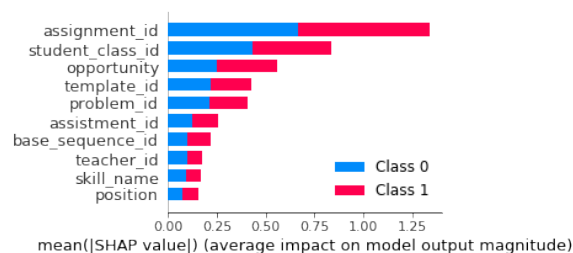


Figure 5: SHAP values for ASSIST09 (meta-data)

This is on par with our previous results, having shown the importance of assistance in predicting correctness. A possible explanation to this high impact on prediction is that questions with built-in scaffolding are likely to be of a higher difficulty level, leading the instructor to include scaffolding questions. Likewise, *original*, indicating a question isn't part of a scaffolding, has a moderately strong impact.

Besides scaffolding, both models rely on contextual information about the questions, such as the ID of the problem set or the ID of the problem itself. In ASSIST09, the additional information about the students' background, represented by their class and teacher IDs, is shown to be important to the predictive ability of the model.

The strong results achieved by these models, with very little information about the user's studies and history of previous answers, highlight the value of the representations the model learned. Without relying on user-success history, this contextual meta-data only model is able to reach a high AUC score, even outperforming the classical BKT approach on ASSIST09. This further reinforces the potential of integrating novel techniques to leverage contextual information when evaluating student mastery rather than relying solely on students' answers history.

6. CONCLUSION AND FUTURE WORK

In this paper, we introduced a novel deep learning model able to efficiently learn deep representations of contextual assessment information.

We showed that the proposed model reaches a very high level of performance when using both meta and instance-specific data on predicting whether a student will correctly answer a question or not.

We further showed that we can reach a relatively high level of performance on the same task while using exclusively contextual meta-data and very limited student-related information.

Additionally, our analysis of the information learned by the model shows that there is valuable insight to be extracted from analysing its predictions.

This work highlights the potential of learning from contextual data on top of user-history data and could be extended in several ways.

Future work should focus on integrating such learned representations within traditional knowledge tracing systems and learning schedulers and comparing their predictions to those of DPE. Contextual information is complementary to the information these systems exploit and could lead to improvements in the learning process. We also intend to investigate how the results we have obtained could be used to enrich

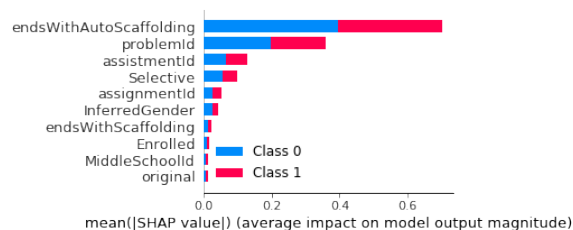


Figure 6: SHAP values for ASSISTChall (meta-data)

theory-grounded models such as DeepIRT [27].

Furthermore, such an approach opens the way to extending current systems with additional external information, such as information about a user's interaction with course materials surrounding the knowledge evaluated.

7. REFERENCES

- [1] The 2017 assistments datamining competition. <https://sites.google.com/view/assistmentsdatamining>.
- [2] O. O. Adesope, D. A. Trevisan, and N. Sundararajan. Rethinking the use of tests: A meta-analysis of practice testing. *Review of Educational Research*, 87(3):659–701, 2017.
- [3] B. Choffin, F. Popineau, Y. Bourda, and J. Vie. DAS3H: modeling student learning and forgetting for optimally scheduling distributed practice of skills. In *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada, July 2-5, 2019*, 2019.
- [4] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4:253–278, 1994.
- [5] M. Feng, N. Heffernan, and K. Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. *User Model. User-Adapt. Interact.*, 19:243–266, 08 2009.
- [6] C. Guo and F. Berkhahn. Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*, 2016.
- [7] R. K. Hambleton, H. Swaminathan, and H. J. Rogers. *Fundamentals of item response theory*. Sage, 1991.
- [8] J. Howard and S. Gugger. Fastai: A layered api for deep learning. *Information*, 11(2):108, Feb 2020.
- [9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15*, page 448–456. JMLR.org, 2015.
- [10] B. Kaur and S.-F. Yap. Kassel project report—third phase. *Singapore: National Institute of Education.*, 1998.
- [11] J. M. Keiser and D. V. Lambdin. The clock is ticking: Time constraint issues in mathematics teaching reform. *The Journal of Educational Research*, 90(1):23–31, 1996.
- [12] Y. Leong and H. Chick. Time pressure and instructional choices when teaching mathematics.

Mathematics Education Research Journal, 23:347–362, 09 2011.

- [13] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.
- [14] D. Misra. Mish: A self regularized non-monotonic neural activation function, 2019.
- [15] M. C. Mozer and R. V. Lindsey. Predicting and improving memory retention: Psychological theory matters in the big data era. In *Big data in cognitive science*, pages 43–73. Psychology Press, 2016.
- [16] S. Pandey and G. Karypis. A self attentive model for knowledge tracing. In *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada, July 2-5, 2019*, 2019.
- [17] Z. A. Pardos and N. T. Heffernan. KT-IDEM: introducing item difficulty to the knowledge tracing model. In J. A. Konstan, R. Conejo, J. Marzo, and N. Oliver, editors, *User Modeling, Adaption and Personalization - 19th International Conference, UMAP 2011, Girona, Spain, July 11-15, 2011. Proceedings*, volume 6787 of *Lecture Notes in Computer Science*, pages 243–254. Springer, 2011.
- [18] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Performance factors analysis—a new alternative to knowledge tracing. *Online Submission*, 2009.
- [19] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 505–513. Curran Associates, Inc., 2015.
- [20] L. Razzaq, M. Feng, G. Nuzzo-Jones, N. Heffernan, K. Koedinger, B. Junker, S. Ritter, A. Knight, C. Aniszczuk, S. Choksey, et al. The assistment project: Blending assessment and assisting. In *Proceedings of the 12th annual conference on artificial intelligence in education*, pages 555–562, 2005.
- [21] L. Razzaq and N. T. Heffernan. Scaffolding vs. hints in the assistment system. In *International Conference on Intelligent Tutoring Systems*, pages 635–644. Springer, 2006.
- [22] S. Reddy, S. Levine, and A. Dragan. Accelerating human learning with deep reinforcement learning. In *NIPS workshop: teaching machines, robots, and humans*, 2017.
- [23] B. Settles and B. Meeder. A trainable spaced repetition model for language learning. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1848–1858, 2016.
- [24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [25] B. Tabibian, U. Upadhyay, A. De, A. Zarezade, B. Schölkopf, and M. Gomez-Rodriguez. Enhancing human learning via spaced repetition optimization. *Proceedings of the National Academy of Sciences*, 116(10):3988–3993, 2019.
- [26] Y. Wang and N. T. Heffernan. The “assistance” model: Leveraging how many hints and attempts a student needs. In *Twenty-Fourth International FLAIRS Conference*, 2011.
- [27] C. Yeung. Deep-irt: Make deep learning based knowledge tracing explainable using item response theory. In *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada, July 2-5, 2019*, 2019.
- [28] C.-K. Yeung and D.-Y. Yeung. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, pages 1–10, 2018.
- [29] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*, 2019.

More Data and Better Keywords Imply Better Educational Transcript Classification?

Theodora Ioana Danciulescu,
Marian Cristian Mihaescu
University of Craiova
Department of Computer Science
theodora_danciulescu@yahoo.com
mihaescu@software.ucv.ro

Stella Heras, Javier Palanca,
Vicente Julian
Universitat Politècnica de Valencia
Sistemas Informáticos y Computación
stehebar@upv.es, jpalanca@dsic.upv.es
vjulian@upv.es

ABSTRACT

Building and especially improving a classification kernel represents a challenging task. The works presented in this paper continue an already developed semi-supervised classification approach that aimed at labelling transcripts from educational videos. We questioned whether the size of the ground-truth data-set (Wikipedia articles) or the quality of the keywords used in the semi-supervised labelling have a significant impact on the accuracy metrics of the final obtained data model. Experimental results took into consideration three Wikipedia data-sets of *Small*, *Medium* and *Large* sizes. For each data-set there were used three sets of keywords: offered by video authors, determined by *rake-nltk* on available transcripts and determined by *rake-nltk* on Wikipedia articles that serve as training and testing data for the LDA model that determine keywords on the transcripts. Experiments show that the size of the data-set has little importance, while the quality of the keywords has a more significant impact. Therefore, an improved version of the previously developed classifier has been obtained by improving the quality of the keywords involved in semi-supervised training. This result paves the way towards further improvements that may finally be deployed as within a recommender system of educational videos at the Universitat Politècnica de València.

Keywords

classification, educational transcripts, keywords, data-set size

1. INTRODUCTION

Over the last few years, the quantity of online learning objects (LO) [6] and Massive Online Learning Courses (MOOCs) have increased dramatically representing a real boom in online learning. This boom of online learning resources has caused a problem for students, as they have hundreds of thousands of online documentation. At the same time, different approaches to discover topics and hidden semantic structures in text have been proposed with the goal of go

forward on topic modelling which has been a challenging and critical issue for information retrieval. Therefore, taking into account all of this, topic modelling has become in a trending topic for the e-learning research community. Following that trend, the *Universitat Politècnica de València* (UPV) in Spain launched a video lectures sharing website, called *Polimedia*¹, and a MOOC platform, called *UPV[X]*², which is powered by the edX MOOC platform³.

Both proposals have a basic search engine allowing students to search for videos (learning objects) by simply using a set of keywords. Current solutions compare these keywords with some typical metadata of the videos (title, authors, ...) and returns the set of videos that match with this data. Obviously, this basic retrieval solution overlooks any semantics, which produces incomplete results that do not take into account some videos that are relevant for the student but that do not include any of the keywords in their titles.

The MOOCs we are using in this work consists of a set of educational videos that have an automatic transcription of the lectures that is going to be used as part of the input data for this proposal. The motivation of this work is to use this information to help students to find more suited learning objects, personalized to their interests, in these massive online platforms where the number of learning objects grows quickly and they usually are not tagged correctly.

According to this, this paper focuses on the improvement of this search engine proposing a new retrieval method that uses a dataset extracted from Wikipedia articles and that is trained to classify keywords based on the topic of the available educational videos. This proposed model is an improvement of a previous work presented in [14], where pre-tagged wikipedia articles were used as ground-truth. In this work we improve this semi-supervised method by: 1) automatically tagging Wikipedia articles and using them to create an extended dataset for training the semi-supervised method, and 2) proposing an improved pipeline for cleaning the data, extract keywords and obtain a better classification model that improves the precision of the student's searches.

The rest of the paper is structured as follows: Section 2 presents some works related to the topic of this paper; Sec-

¹UPV Media, <https://media.upv.es>

²UPV[X], <https://www.upvx.es/>

³edX MOOC platform, <https://www.edx.org/>

tion 3 details the approach proposed by the authors; Section 4 presents some experimental results; and finally, Section 5 shows the conclusions of this work.

2. RELATED WORK

The problem of the correct keyword extraction is a recurrent problem over the last few years. Different works have appeared trying to solve this problem using different approaches. At the end, the idea is to have a solid set of words that concisely represent the content of a text (in this case the content of a learning object).

Most of the last approaches on document-oriented methods of keyword extraction use natural language processing (NLP) techniques mainly based on machine learning algorithms and statistical methods. One of the most well-known approaches is the work presented in [17] where authors propose the use of Support Vector Machines as a way to extract the most important keywords.

On the other hand, the work in [9] presents a solution based on the graph-based syntactic representation of text and web documents that combines supervised and unsupervised learning. In a similar way, the work presented in [7] proposes an unsupervised keyword extraction technique including several different ways of the conventional TF-IDF model with reasonable heuristics. Other approaches, like the work presented in [12] called Rapid Automatic Keyword Extraction (RAKE), employ unsupervised methods for extracting keywords which are domain-independent, and also, language-independent.

The latent Dirichlet allocation (LDA) model is one of the most used techniques to classify documents according to a set of topics. One example is the work presented in [1] that automatically captures the thematic patterns and identifies emerging topics using a non-Markov on-line LDA Gibbs sampler topic model. In the online educational field, the LDA model has been used in works such as the presented in [16] where the authors use topic detection for the analysis of the feedback submitted by students in online courses. The work in [10] tries to solve the problem of topic detection by identifying words that appear with high frequency in the topic and low frequency in other topics.

Some works face the keyword extraction problem in learning objects through the use of other approaches such as ontologies like the work presented in [8] that aims to improve the effectiveness of retrieval and accessibility of learning objects integrating semantic knowledge through domain-specific ontologies. In [4] authors use Wikipedia to associate learning objects to Wikipedia pages, specifically with the topics of those pages, trying to find relationships among learning objects.

Finally, recent work also uses intelligent algorithms and method to face other challenges of efficient videolectures management, such as video shots skimming [15] and supervised multi-class classification [5].

Opposite to most related works, our method is fully semi-supervised, with no need for a previously tagged database nor an ontology, that can act as ground truth to train the

models. Also, to the best of our knowledge, there are no other intelligent systems trained to automatically classify a Spanish database of educational videos.

3. PROPOSED APPROACH

From a classification perspective, the first issue is to clearly state the actual number of topics (i.e., labels) that exist in available transcripts. Since all transcripts come from educational videos from UPV, it certainly means that the number of topics is represented by the domains from which videos come from, that is *biology & sciences* (BS), *engineering* (E) and *humanities & arts* (HA). BS topic considers aspects of bacteria, diseases, bio-engineering, bio-medicine, E topic considers aspects of computers, electrical, architecture, civil, aerospace. In contrast, HA considers aspects of laws, arts, social and economic.

The proposed approach extends the semi-supervised method described in a previous paper-work[14]. It improves the data analysis pipeline in terms of accuracy of classification on the videos currently available in the database. As in the initial approach, the training on Wikipedia articles uses the SVM[3] classification algorithm, which used a Radial Basis Function (RBF) kernel from the *sklearn* library[11]. The validation approach uses the same two steps: 1) train on 70% of Wikipedia articles and cross-validate with 15%, 2) train on labelled transcripts and validate on remaining unseen 15% of Wikipedia articles.

Internally, the semi-supervised training has been performed on a set of labelled Wikipedia articles by building a data model that has been used for classifying educational transcripts and their associated keywords. The transcripts which had the same label as the keywords were considered correctly labelled and therefore were added to the initial training dataset. The newly obtained dataset is used in an iterative semi-supervised set up for training in an attempt to tag as many educational transcripts as possible.

One limitation of previous works is that HA items were mislabeled as E. This flaw may be caused by the fact that videos about HA reach more various subjects, that are not so domain-specific. Mathematics videos with proofs demonstration and analysis are also not correctly labelled as there is a large number of words that are not mathematics domain-specific. Many videos about the economy and economic environments tend to be categorised as E, as many explanations heavily use mathematics and calculus. A positive aspect is that the classification for BS items is acquiring excellent results, there are no confusions made for this domain. This behaviour is expected as this domain has many specific terms and principles, so videos from this area are easily classifiable and do not create confusions.

As a first step to improve the previous work[14] was to extend the Wikipedia articles data-set for training the semi-supervised method. This was done progressively, as we compared results with the previous ones and checked manually if the videos that were badly classified have been classified correctly. The decision about the amount and about which Wikipedia articles categories should be downloaded was made by manually analysing the clustering results from previous work. By doing so, we obtained best results with

three versions of datasets: a *Small* data-set (3747 Wikipedia articles), a *Medium* dataset (6373 Wikipedia articles) and a *Large* dataset (18527 Wikipedia articles).

Secondly, we focused on the importance of relevant keywords to obtain a good classification result. There were provided three sets of keywords supplied by three different methods. The first set was obtained using the same process from the previous paper[14] by using the keywords provided by the videos' authors. However, we observed inconsistencies as some videos do not offer keywords in their metadata. The second set of keywords was obtained by using *rake-nltk*[13] tool for extracting the keywords directly from the transcripts' text. Finally, the third set was obtained by using *rake-nltk* tool for getting keywords from the *Large* data-set of Wikipedia articles to use them as training and testing data for an LDA (Latent Dirichlet Allocation)[2] model that will extract domain-specific tags from the transcripts.

3.1 Training on more Wikipedia articles

Intuitively, more data should help to improve the accuracy, but in practical situation this may not happen. An issue that currently occurs in machine learning systems is whether or not the size of the data-set is too small for the classification problem. Proper debugging of the data analysis pipeline should clearly point out if current accuracy results may be improved by using a larger data-set or other leverages should be taken into consideration.

As a first approach, we tried to detect a pattern in the classification errors and download the appropriate Wikipedia articles to cover the subjects in the videos that were mistakenly classified. Consequently, when choosing the Wikipedia articles, not only the covering of the topics was taken into consideration but also the quantity of the articles about that subject was an important factor.

In response to this, additionally to the initial *Small* dataset used in the previous work[14] we obtained two new datasets: *Medium* dataset with a total of 6373 articles (i.e., 1219 BS articles, 2737 HA articles, and 1626 E articles), and a *Large* dataset with 18526 Wikipedia articles (including 5830 BS articles, 5882 E articles and 6814 HA articles).

3.2 Determining better keywords

The transcripts' keywords represent a key-point for the classification algorithm, as the quality of the classification may be directly influenced by the relevance and quality of the keywords.

A second solution was represented by the *rake-nltk* tool, as it supports the Spanish language and it provides good results for this language, too. *Rake-nltk* tool is a domain-independent keyword extraction algorithm which tries to determine key phrases in a body of text by analyzing the frequency of word appearance and its co-occurrence with other words in the text.

After trying to classify the videos in 3 clusters (BS, E and HA) using three different sized data-sets (i.e., *Small*, *Medium* and *Large*) for training and two different methods for assigning keywords to each transcript (the manually provided keywords by authors and the keywords extracted with *rake-*

nltk), we finally use the third method of providing more domain-specific keywords for every transcript: we used LDA as business logic for the implementation of transcript keywords recommendation system and used *rake-nltk* for providing keywords for Wikipedia articles to obtain training and testing data.

As the transcripts and the keywords from the metadata (i.e. authors' keywords) do not represent a valid data-set (the words used as keywords are either ambiguous, either too name specific and they often induce classification errors).

The limitation of the second method consists from the fact that the keywords provided by *rake-nltk* from transcripts were large and with numerous phrases without a focus on the essential subject of the video, also causing classification errors in some cases. So, a third solution was needed: there were used Wikipedia articles and keywords extracted with *rake-nltk* as training and testing data set for the LDA model to extract domain-specific keywords from the transcripts. The third solution is combining the *rake-nltk* tool with the LDA model. *Rake-nltk* will be used to extract keywords from the Wikipedia articles resulting in a labelled dataset that will serve later as training and testing dataset for the LDA model to extract domain-specific keywords from the transcripts.

The second approach provides new keywords for every transcript by using *rake-nltk*. The keywords extracted with this tool were also pre-processed by eliminating stop words and lowering all the letters. However, there still is one disadvantage for this method: the keywords extracted are large phrases that are not necessarily very domain-specific. Moreover, the extracted sentences are ambiguous in some cases, lacking the essential subject of the transcript. This error is most likely to be caused by the fact that the transcripts are not always subject-focused, they usually have an introduction about the teacher, the subject in general, many examples are provided. Hence, there is a broad set of words that may induce errors.

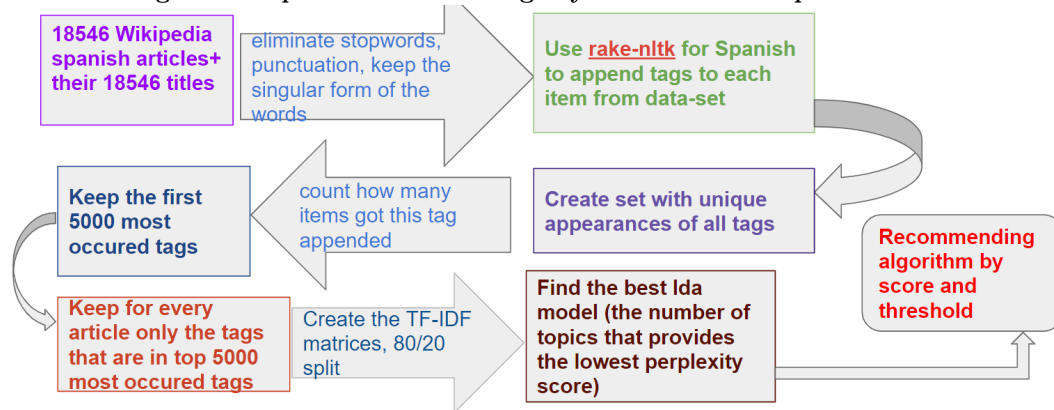
The third approach used *rake-nltk* tool, not for extracting keywords directly for our transcripts, but for extracting keywords for each article from the Wikipedia articles *Large* data-set (18526 articles). The tagged Wikipedia articles using *rake-nltk* will be used as training data for assigning keywords to the video transcripts employing LDA.

Figure 1 presents in detail the data analysis pipeline for the third method of providing keywords. This method is being described in this section in particular.

The following steps were followed for obtaining the domain-specific transcript tags recommendation algorithm utilizing LDA:

Create a balanced and large data-set of Wikipedia articles in Spanish. By saying to have a balanced data-set, there are supposed to be enough BS articles to obtain a set of keywords for BS, enough E articles to get a set of keywords for this domain, and most important enough HA articles to form a set of tags for this domain, too. The difficult part was to get a good set of keywords for HA domain, as this cluster covers a wide range of fields like Economy, Law,

Figure 1: Pipeline for extracting keywords from Wikipedia articles



Arts, Architecture, Language learning, Politics, Social Sciences, Philosophy, Psychology and basically anything that does not fit in the other two clusters.

Clean the text from the downloaded Wikipedia articles by lowering text, removing undesirable marks and stop words, using the singular form of the word. Append each Wikipedia article tags using *rake-nltk* tool and also *clean* (lower text, remove undesirable marks, remove stop words, use the singular form of the word) these tags. For better results, there are also tags extracted from the titles of the Wikipedia articles. That means that we pull tags for 18526 x 2 items.

Add all these tags in a set to have only unique appearances of the extracted tags.

Count how many Wikipedia articles were assigned to each tag from the set.

Get top 5000 most occurred tags (having less tags, it means that only the most occurred tags from each domain will be kept, and in this way, a classification with the semi-supervised method will be simpler to perform with a smaller training data-set)

Keep only the top 5000 occurring tags for each Wikipedia article.

Keep only the articles that are still labelled. After these operations, we end up with 21743 labelled items out of 37092 items.

Create the TF-IDF matrices by splitting our obtained data set in 80%/20%.

We try to **train various LDA models** using *sklearn*⁴ implementation [18], by assigning each of them a different topic number, then the different models are evaluated on the test set using the metric perplexity. By definition, the lower the perplexity, the better the model.

Showing the perplexity score for several LDA models with different values for *n_components* parameter, and printing the top words for the best LDA model (the one with the lowest perplexity).

Now that we have designed the workflow, we focus on the keywords recommendation algorithm for the transcripts, which is based on two main aspects:

- **Score** = probability that document is assigned to a specific topic, represents the topic's probability of generating the word.
- A word is considered as a relevant tag, when its score is superior to a defined threshold. After testing different values for the threshold, we decided to choose the threshold to 0.008, that is because, for this value, because with a threshold equals to 0.008 more than 95 percents of the transcripts have recommended tags.

Also, an advantage for obtaining keywords for every transcript employing *rake-nltk* combined with LDA would be that all the videos will be classified. In the original method, only the videos that were provided keywords by authors could have been taken into consideration. Now, as we offer keywords to every transcript, all the videos with an available transcript may be taken into consideration. An even bigger advantage is the fact that the training set contains articles about well-defined domains, their subject is focused on a small range of ideas, so the set of most frequently used tags will be very domain-specific, a fact that will be helpful for the classification algorithm.

4. EXPERIMENTAL RESULTS

After running the semi-supervised learning method for the *Small*, *Medium* and *Large* data-sets, and also with the three sets of keywords, the best results were obtained by training the semi-supervised method with the *Small* data-set of Wikipedia articles and the keywords provided employing *rake-nltk* for obtaining training and testing data and LDA to obtain the proper transcript's tags. The results are presented in Table 1. This table also provides a detailed insight of the semi-supervised training process results along with the number of transcripts added to the model in every iteration and with the classification accuracy obtained for each label. The computation of the classification accuracy metrics is done on the validation data-set, which contains only unseen data in the training step.

Analysis of the iterative semi-supervised training process in all nine scenarios (i.e., for three data-set sizes and for three methods of obtaining the keywords) revealed several pat-

⁴<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html>

Iteration (valid /available)	Accuracy	Class	Precision	Recall	F1-score
#1 (8487 / 14395)	0.92 (+/- 0.01)	Biology&Sciences	0.95	0.93	0.94
		Engineering	0.88	0.92	0.90
		Humanities&Arts	0.95	0.93	0.94
#2 (2375 / 5908)	0.96 (+/- 0.02)	Biology&Sciences	0.92	0.94	0.93
		Engineering	0.87	0.88	0.87
		Humanities&Arts	0.95	0.93	0.94
...
#8 (9 / 1940)	0.94 (+/- 0.05)	Biology&Sciences	0.90	0.94	0.92
		Engineering	0.85	0.86	0.85
		Humanities&Arts	0.93	0.90	0.92

Table 1: Validation scores for each iteration in the pipeline with the *Small* Wikipedia articles data-set and the keywords provided by means of *rake-nltk* for obtaining training and testing data and LDA to obtain the proper transcript’s tags.

terns. The first observation regards the fact that the number of iterations has low variance. So, irrespective of the size of Wikipedia data-set and the method for obtaining the keywords the number of iterations is in the range from six to twelve. This observation represents a clear indication that the size of the training data-set of Wikipedia articles does not highly influence the semi-supervised learning. Another observation is that each step in the semi-supervised training keeps unchanged or slightly decreases the F1 score, while slightly increasing the accuracy of the 10-fold cross-validation on the Wikipedia test data-set. This observation shows that all experiments are consistent and produce similar behavioural patterns in terms of accuracy, precision, recall and F1-score measures evolution in terms of evolution during semi-supervised training.

Table 2 presents the validation scores for all the three data-sets (i.e., *Small*, *Medium* and *Large*) and all three keywords data-sets.

The first observation regarding the validation results from table 2 regards the fact that there are no big differences in terms of overall accuracy and F1-scores for the three data-sets of keywords and for each training data-set. Still, the method with *rake-nltk* for Wikipedia articles keywords and LDA for obtaining transcript keywords generally has better scores than the other two methods for cluster 2. Still, it has usually lower scores for cluster 1. This pattern shows an indication that improvements in classification metrics should focus on classes where poorly results occur.

We further observe that scores tend to slightly decrease as the data-set is getting larger. Therefore for the *Medium* data-set, only the method with *rake-nltk* for extracting transcript keywords provides better results than it does with the *Small* data-set. A particular result consists in major score decreases for cluster 1 for the *Medium* data-set. This is mainly due to the unbalance of this data-set regarding the items from labelled in class 1. The imbalance of class 1 is also signalled by the excellent results for classes 0 and 1 in the experiment with *Large* data-set and the method with *rake-nltk* for Wikipedia articles keywords and LDA for transcript keywords.

Despite the *Large* data-set used for training the model, comparing the time required to train the model with the *Small*

data-set and the time necessary to train the model with the *Large* data-set with all three sets of keywords, we have noticed that the time has doubled in the worst case, even though the data-set used is 6 times larger than the initial one.

Besides, the method to obtain keywords employing *rake-nltk* and LDA transcript keywords provide a better running-time execution for the *Small* and *Large* data-sets than the original keywords set as the number of iterations is also smaller.

The method with *rake-nltk* and LDA transcript keywords provides best result for the *Small* data-set, though the *rake-nltk* transcript keywords methods has the best results for the *Medium* and *Large* data-sets. For the method to obtain domain-specific keywords for transcripts employing *rake-nltk* to extract Wikipedia articles keywords and LDA to extract the proper keywords for transcripts, the tags distribution per the 10 topics of the model is presented in Table 3. We also notice that the 10 topics do not mix the three domains that we are interested about: E tags are found only in topics that do not contain tags from the other two domains, and the same for BS tags and HA tags. There can be easily noticed the domain that each topic covers: the topics with indexes 1, 2, 5, 9 and 10 are focused on HA domain, the topics with indexes 4, 6 and 8 are focused on E domain, and finally, the topics 3 and 7 are focused on BS domain.

Furthermore, the topic order shows that the first three most important topics are 4, 3 and 9, where 4 is focused on the E domain, 3 is concentrated in BS tags, and 9 is focused on HA tags. Considering that the first three most important topics contain one topic for each of the three domains that we are interested in, ultimately confirms that the model is suitable for our purpose. In addition, the following 3 topics in the topic order are also distributed equally across the three domains.

We can notice that the original keywords provided by authors are provided in different styles: some of them are too specific(tool names that are not so common), some of them too ambiguous to be categorised to a domain, and some of them provide domain-specific terms, but those terms may not be so standard in that domain in such a way to be correctly categorised by put semi-supervised method that is not trained on a massive data-set.

Table 2: Validation scores for all data-sets and keywords sets

Data-set	Keywords	Accuracy/Avg F1	Class	Precision	Recall	F1-score
Small	Original keywords	0.94/0.88	0	0.96	0.86	0.91
			1	0.84	0.86	0.85
			2	0.86	0.90	0.88
	<i>rake-nltk</i> transcript keywords	0.94/0.88	0	0.96	0.86	0.91
			1	0.84	0.87	0.86
			2	0.86	0.90	0.88
	<i>rake-nltk</i> and LDA transcript keywords	0.95/0.89	0	0.91	0.92	0.91
			1	0.81	0.91	0.86
			2	0.93	0.84	0.88
Medium	Original keywords	0.94/0.86	0	0.93	0.83	0.88
			1	0.75	0.92	0.83
			2	0.93	0.82	0.87
	<i>rake-nltk</i> transcript keywords	0.96/0.88	0	0.93	0.85	0.89
			1	0.80	0.90	0.85
			2	0.93	0.87	0.90
	<i>rake-nltk</i> and LDA transcript keywords	0.94/0.85	0	0.93	0.84	0.88
			1	0.72	0.91	0.80
			2	0.93	0.79	0.85
Large	Original keywords	0.95/0.86	0	0.95	0.82	0.88
			1	0.80	0.90	0.85
			2	0.86	0.85	0.85
	<i>rake-nltk</i> transcript keywords	0.96/0.86	0	0.95	0.81	0.88
			1	0.82	0.88	0.85
			2	0.83	0.87	0.85
	<i>rake-nltk</i> and LDA transcript keywords	0.95/0.85	0	0.93	0.82	0.87
			1	0.77	0.90	0.83
			2	0.86	0.82	0.84

Table 3: Highest score tags per topics in the LDA model

T 1	derecho / social / sociedad / política / cultura
T 2	dato / software / aplicación / versión / código
T 3	célula / proteína / agua / animal / forma / celular
T 4	algoritmo / error / programa / memoria / ejecución
T 5	mercado / precio / economía / financiero / empresa
T 6	displaystyle / teoría / lógica / matemática
T 7	tratamiento / cirugía / médico / paciente / síndrome
T 8	ecuación / ingeniería / inteligencia / artificial
T 9	política / análisis / marketing / rama / arteria
T 10	industrial / industria / plano / internacional
Order	[4, 3, 9, 8, 7, 5, 10, 6, 2, 1]

The third method, the one that uses *rake-nltk* for providing keywords to the Wikipedia articles used for training and LDA for extracting transcript tags, provides a few labels, but they are very domain-specific. The tags that can be resulted from this method come from a relatively small set of possible tags (this set is formed by the most commonly used terms in the 3 domains of our clusters), so the most relevant tags from this set will be chosen.

This is an advantage for our semi-supervised method as we can provide good results with a relatively small data-set for training. The words used for tags by this method are very likely to be well categorised by the semi-supervised method as they are very common only in the are of one of the three domains.

5. CONCLUSIONS

This paper has presented a method which combines the extraction of keywords from a Wikipedia data-set with the automatic classification of learning objects using LDA to obtain better keywords for searching educational videos. This will allow students to find more accurate resources for videos that have not been appropriately tagged by authors.

Using Wikipedia for creating a labelled data-set has allowed us to build a balanced set of articles that have been used to train a model for extracting keywords from educational video transcripts. However, in future works, it would be interesting to provide an automatic mechanism for building balanced training data-sets.

The proposed has been tested using a real environment, concretely the video lectures sharing website of the *Universitat Politècnica de València*, which has more than 55.000 short videos mainly in Spanish. Results have shown the benefits of this proposal for classifying learning objects into categories (specifically Biology&Sciences, Engineering and Humanities&Arts), which will help students in their search of appropriated learning resources.

Future works should focus on improving accuracy of the classification especially for the classes with poorer results, that is *Engineering* and *Humanities & arts* as *Biology* transcripts are correctly classified. The obtained classifier may be further used for labeling new videos that may be added into UPV Media site.

6. REFERENCES

- [1] L. AlSumait, D. Barbará, and C. Domeniconi. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *2008 eighth IEEE international conference on data mining*, pages 3–12. IEEE, 2008.
- [2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Lda(latent dirichlet allocation). In *Advances in neural information processing systems*, pages 601–608, 2002.
- [3] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [4] C. De Medio, F. Gasparetti, C. Limongelli, F. Sciarone, and M. Temperini. Automatic extraction of prerequisites among learning objects using wikipedia-based content analysis. In *International conference on intelligent tutoring systems*, pages 375–381. Springer, 2016.
- [5] D. Dessì, G. Fenu, M. Marras, and D. R. Recupero. Leveraging cognitive computing for multi-class classification of e-learning videos. In *European Semantic Web Conference*, pages 21–25. Springer, 2017.
- [6] S. Downes. Learning objects: resources for distance education worldwide. *The International Review of Research in Open and Distributed Learning*, 2(1), 2001.
- [7] S. Lee and H.-j. Kim. News keyword extraction for topic tracking. In *2008 Fourth International Conference on Networked Computing and Advanced Information Management*, volume 2, pages 554–559. IEEE, 2008.
- [8] L. Lemnitzer, C. Vertan, A. Killing, K. Simov, D. Evans, D. Cristea, and P. Monachesi. Improving the search for learning objects with keywords and ontologies. In *European Conference on Technology Enhanced Learning*, pages 202–216. Springer, 2007.
- [9] M. Litvak and M. Last. Graph-based keyword extraction for single-document summarization. In *Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization*, pages 17–24. Association for Computational Linguistics, 2008.
- [10] T. Liu, N. L. Zhang, and P. Chen. Hierarchical latent tree analysis for topic detection. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 256–272. Springer, 2014.
- [11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. sklearn. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.
- [12] S. Rose, D. Engel, N. Cramer, and W. Cowley. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20, 2010.
- [13] V. B. Sharma. Rapid automatic keyword extraction algorithm using nltk. <https://pypi.org/project/rake-nltk>, 2019.
- [14] A. S. Stoica, S. Heras, J. Palanca, V. Julian, and M. C. Mihaescu. A semi-supervised method to classify educational videos. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 218–228. Springer, 2019.
- [15] B. N. Subudhi, T. Veerakumar, S. Esakkirajan, and S. Chaudhury. Automatic lecture video skimming using shot categorization and contrast based features. *Expert Systems with Applications*, page 113341, 2020.
- [16] S. Unankard and W. Nadee. Topic detection for online course feedback using lda. In *International Symposium on Emerging Technologies for Education*, pages 133–142. Springer, 2019.
- [17] K. Zhang, H. Xu, J. Tang, and J. Li. Keyword extraction using support vector machine. In *international conference on web-age information management*, pages 85–96. Springer, 2006.
- [18] G. Zhao, Y. Liu, W. Zhang, and Y. Wang. sklearn.decomposition.latentdirichletallocation. In *Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences*, pages 188–191, 2018.

Zero-shot Learning of Hint Policy via Reinforcement Learning and Program Synthesis

Aleksandr Efremov
MPI-SWS
aefremov@mpi-sws.org

Ahana Ghosh
MPI-SWS
gahana@mpi-sws.org

Adish Singla
MPI-SWS
adishs@mpi-sws.org

ABSTRACT

Intelligent tutoring systems for programming education can support students by providing personalized feedback when a student is stuck in a coding task. We study the problem of designing a *hint policy* to provide a next-step hint to students from their current partial solution, e.g., which line of code should be edited next. The state of the art techniques for designing a hint policy use supervised learning approach, however, require access to historical student data containing trajectories of partial solutions written when solving the task successfully. These techniques are limited in applicability when needed to provide feedback for a new task without any available data, or to a new student whose trajectory of partial solutions is very different from that seen in historical data. To this end, we tackle the *zero-shot* challenge of learning a hint policy to be able to assist the very first student who is solving a task, without relying on any data. We propose a novel *reinforcement learning* (RL) framework to solve the challenge by leveraging recent advancements in RL-based neural *program synthesis*. Our framework is modular and amenable to several extensions, such as designing appropriate reward functions for adding a desired feature in the type of provided hints and allowing to incorporate student data from the same or related tasks to further boost the performance of the hint policy. We demonstrate the effectiveness of our RL-based hint policy on a publicly available dataset from Code.org, the world's largest programming education platform.

1. INTRODUCTION

In recent years, there has been an increasing focus on developing educational tools for STEM (science, technology, engineering, and mathematics) and computing. Problem-solving skill, i.e., ability to solve multi-step problems by deductive reasoning, is one of the key ingredient of learning in these domains [14, 25]. For instance, while working on a coding task, a student iteratively writes, tests and refines the code to arrive at the final solution [8, 23, 27, 24].

One of the difficulties in designing assistive algorithms for these open-ended coding tasks is that the state space, i.e., the set of partial solutions that students might arrive at when solving the task, is *unbounded*. For instance, for a simple coding task from the *Hour of Code* (HOC) challenge by Code.org [5], the correct solution contains only 5 blocks (see Figure 1), whereas students can create millions of unique partial solutions in the process of solving the task [23]. When solving such tasks, it is evident that students can get stuck at a state (i.e., a partial solution) and do not know how to proceed (i.e., which action/edit to apply). Intelligent tutoring systems empowered by machine learning techniques held a great promise in supporting such stuck students by providing personalized feedback, e.g., explaining misconceptions and giving guidance on what to do next [31, 16, 2, 24].

We focus on the well-studied feedback mechanism in programming education called *next-step* hints: When a student is stuck at a given state, the system suggests the next edit that student should make to their current code to proceed [3, 8, 16, 23, 27, 20]. In the context of block-based languages that are extremely popular in educational tools for visual programming [21, 5, 24], the suggested hints correspond to one of the allowed actions from the student's current code (e.g., adding or removing a block, and changing a conditional in one of the blocks), see Figure 1. Inspired by the work of [3, 23, 20], we refer to the function that provides such hints to the student from any partial solution as *hint policy*.

The key challenge in designing a hint policy is that the space of partial solutions is unbounded even for simple coding tasks and there is a huge variability in students' trajectories of partial solutions [23, 20, 34]. A number of techniques proposed in the literature use a graph representation of the task (with nodes denoting partial solutions and edges denoting single edits that convert one partial solution to another) [3, 9, 23, 35, 27]. These techniques then use historical student data and domain knowledge to capture the editing behavior of capable students (or experts) on this graph. However, these techniques face serious scaling issues as the problem size grows and are only applicable in settings where we have access to large volume of historic data for the task.

In recent years, new techniques have been developed using a supervised learning approach. These techniques leverage code embeddings to compactly represent the space of partial solutions and can provide hints to students with trajectories that have never been observed in the historical data [22, 20].

Aleksandr Efremov, Ahana Ghosh and Adish Singla "Zero-shot Learning of Hint Policy via Reinforcement Learning and Program Synthesis" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 388 - 394

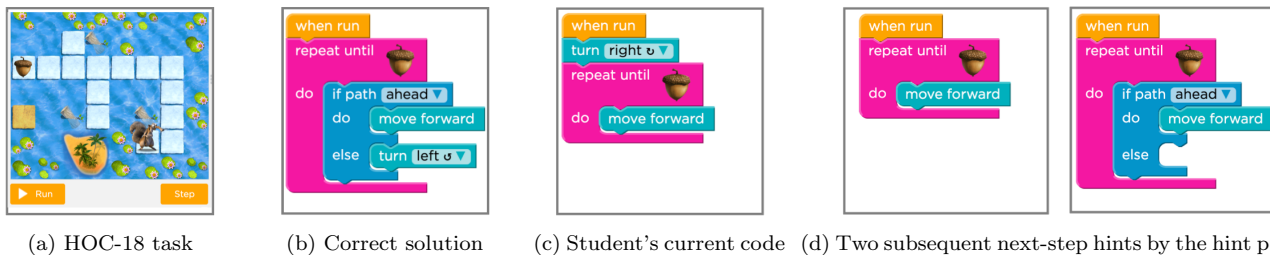


Figure 1: Illustration of next-step hints feedback by our hint policy. (a) shows the HOC-18 task from the *Hour of Code* (HOC) challenge by Code.org [5]. A student solves the task by starting from empty code and builds up the solution using blocks available in the visual interface, also see [23]. (b) shows the correct solution—this is the code that solves the task with minimal number of 5 blocks. (c) shows the current partial solution of a student who is solving this task. (d.left) shows the next-step hint by our hint policy that will be provided as feedback to the student. (d.right) shows the subsequent next-step hint by the policy if student were to ask another hint after receiving the first hint.

In particular, the state of the art technique by [20], *Continuous Hint Factory* (CHF), learns a *regression function* as the hint policy which can identify the most likely hint as a vector in an embedding space and then translates this vector back into a human-readable edit. In comparison to techniques using graph representations, CHF is more scalable and requires access to fewer samples of student data (just enough to learn the generic editing behavior of capable students or experts for the task).

While these state of the art supervised learning techniques are less data-hungry and computationally more powerful, they are still limited in applicability when needed to provide feedback for a new task without any available data, or to a new student whose trajectory of partial solutions is very different from that seen in historical data. Especially with intelligent tutoring systems having the ability to generate tasks on the fly [28, 1, 12], the problem of providing feedback to the very first student on a task is increasingly important. In this paper, we tackle the following *zero-shot* learning challenge: *Can we design a hint policy for a task to provide hints to the very first student solving the task?*

1.1 Our Approach and Contributions

Our approach towards zero-shot learning of hint policy is based on the *reinforcement learning* (RL) framework. In the RL terminology, the set of all possible partial solutions corresponds to the state space, the possible edits from a partial solution defines the state-dependent actions and transition dynamics, and reaching the correct solution quickly yields higher reward (we refer the reader to [26, 29] for a background on RL). Our framework is inspired by recent works [4, 11] that have shown that deep-RL techniques applied to neural embeddings of the code are effective in learning policies to synthesize new programs and to do program repair even if no/minimal training data is available for the task. Intuitively, the problem of providing a hint from a current partial solution is equivalent to one-step of synthesizing the program from this partial solution [17, 10]. However, we note that learning hint policy using RL poses its own practical challenges because the policy needs to provide hints from *any* partial solution which could be arbitrarily bad—this is in contrast to program synthesis and program repair where the initial starting states for RL are limited to either an empty code [4] or a set of partial solutions which are close to the correct solution [11], respectively. The idea of using RL for designing hint policy is also inspired by the seminal

work on *Hint Factory* [3]; however, unlike [3] which relies on historical student data and uses the graph representation of partial solutions, our RL framework uses code embedding and a neural network policy for efficient training.

One might ask what are the advantages of using RL compared to supervised learning techniques for zero-shot challenge. First and foremost, RL enables an effective self-exploration of the solution space by leveraging reward signals (such as receiving higher rewards when a policy can synthesize the correct solution in a fewer steps or can reduce compiler errors). Furthermore, the RL framework is amenable to several extensions for boosting the performance. For instance, if additional student data is available from the same or related tasks, it is possible to bootstrap by combining techniques from imitation learning within RL framework [19, 11]. Also, we can easily incorporate additional human knowledge or features into the policy by designing appropriate reward functions [4, 11]. In summary, this power and flexibility of the RL framework makes it especially suitable for zero-shot learning as it gives us the following ingredients: (i) automatically exploring the solution space or generating synthetic training data [32, 11, 34], (ii) incorporating any available data or expert knowledge to bootstrap and boost the performance [15, 34], and (iii) transferring knowledge from one task to another [18, 6, 7]. Below, we summarize our main contributions:

- We introduce the zero-shot challenge for learning a hint policy to provide next-step hints to the very first student working on a coding task.
- We propose RL framework for zero-shot learning of hint policy. Our framework leverages the representation power of code embedding and a neural network policy for efficiently learning to provide hints. The framework is amenable to several important extensions, e.g., bootstrapping via additional data if available.
- We evaluate the performance of our RL-based hint policy on a publicly available dataset from Code.org, the world’s largest programming education platform [5, 23]. We show significant improvements in next-step hint accuracy w.r.t. the state of the art supervised learning technique.

2. PROBLEM FORMULATION

In this section, we formalize the problem of learning next-step hint policy for programming education.

2.1 Coding Task, Partial Solutions, and Edits

We define the problem in the context of a fixed coding task (e.g., HOC-18 task as shown in Figure 1a). We assume that the correct solution for the task is known. For brevity of presentation, we consider that the correct solution is unique (in fact, the uniqueness holds for HOC-18 task, see Figure 1b). We denote all possible partial solutions for the task by the set S . Note that S is a countable, infinite set. A partial solution $s \in S$ is a piece of code, e.g., as shown in Figure 1, and we denote the correct solution by $s^* \in S$. For any $s \in S$, we define the set of edits that can be applied to s by the action set A_s . In block-based languages, the set A_s corresponds to adding or removing a block in s , editing a conditional for one of the blocks in s , or moving blocks within s . For a partial solution $s \in S$ and an edit $a \in A_s$, the next partial solution obtained by applying a to s is denoted as $s \oplus a$.

2.2 Hint Policy for Next-step Edits

When a student attempts the task, they generate a trajectory of partial solutions denoted as $\xi = (s_0, s_1, s_2, \dots, s_k)$ where k is the trajectory length. Here, s_0 is the empty code, and s_k is the student's latest/current partial solution. Upon reaching s_k , the student might be stuck and is unable to decide how to proceed. Our goal is to help this student by providing feedback as the next-step hint $a \in A_{s_k}$ in the form of an edit that allows the student to make progress. Figure 1c shows one such partial solution s_k and Figure 1d (left) shows the next-step hint that could be provided.

Formally, the next-step hint policy $\pi(\cdot | \xi)$ provides a probability distribution with support over actions A_{s_k} where s_k is the last partial solution in the trajectory ξ . Note that when a policy depends on the whole trajectory ξ , then it can infer the knowledge of the student based on this trajectory and can provide personalized hints. However, the existing hint policy techniques discussed in Section 1 consider myopic policies. A myopic policy $\pi(\cdot | s_k)$ provides a probability distribution with support over actions A_{s_k} and takes as argument only the last partial solution s_k (i.e., ignoring the trajectory of how student reached s_k). In our work, we also focus on learning such a myopic hint policy.

2.3 Evaluation Criteria

As an evaluation criterion, we use the standard approach in literature (e.g., see [23, 20]) where the performance of a hint policy is measured in terms of prediction accuracy. We assume access to a set of expert annotations given by $D_{\text{hints}} = \{(s_i, N(s_i))\}_{i=1,2,\dots,n}$: here, for a partial solution $s_i \in S$, the experts have annotated that the next partial solution where a student should transition to should be among the set $N(s_i) \subseteq S$. In our experiments, we will use the publicly available annotation dataset from [23] for evaluating hint policy on HOC tasks. For a policy π , we use the following notion of *unweighted* prediction accuracy:

$$\frac{1}{n} \sum_{i=1}^n \left(\sum_{a \in A_{s_i}} \pi(a | s_i) \cdot \mathbb{1}(s_i \oplus a \in N(s_i)) \right) \quad (1)$$

where $\mathbb{1}(\cdot)$ represents an indicator function (cf. [23] which uses a notion of accuracy *weighted* by frequency). Note that, this measure of prediction accuracy does not capture the long-term pedagogical value of providing hints to students, and we further discuss this as future work in Section 5.

3. LEARNING HINT POLICY USING RL

In this section, we present our approach to zero-shot learning of hint policy via reinforcement learning (RL) framework.

3.1 RL Framework

3.1.1 Hint policy learning environment as an MDP

In reinforcement learning, a learning algorithm (agent) interacts with an environment typically modelled as a Markov Decision Process (MDP). Here, we present the MDP corresponding to the problem of learning hint policy. We define the MDP $M = (S, A, P, R, S_0)$ as follows:

- S corresponds to the set of partial solutions;
- $A = \cup_{s \in S} A_s$ is the set of all possible actions, and A_s is the set of actions or edits possible in state s ;
- $P : S \times A \times S \rightarrow \mathbb{R}$ denotes the transition dynamics. $P(s' | s, a)$ is defined only for $a \in A_s$. We have $P(s' | s, a) = 1$ for $s' = s \oplus a$, and 0 otherwise.
- $R : S \times A \rightarrow \mathbb{R}$ denotes the reward function. $R(s, a)$ is defined only for $a \in A_s$. A simple reward function could be to set a small negative reward for every action taken and a high reward for reaching the correct solution termed as “goal” (i.e., when $s \oplus a = s^*$). We will discuss more about designing rewards in Section 3.3.
- $S_0 \subseteq S$ is the set of initial states. This corresponds to the states which would be used to initialize an episode when training the hint policy. One way to pick set S_0 is to randomly sample states from S , limited to some upper limit on the code size.

We consider an episodic, finite horizon learning setting [29, 26]: A learning episode starts with an initial state s_0 sampled at random from the set S_0 , then the agent interacts with the environment over discrete time steps t , and the episode ends when one of the following happens: (i) either the agent reaches goal state s^* , or the episode length exceeds a pre-specified threshold (set to 20 in our experiments).

3.1.2 Policy gradient methods

While a variety of RL algorithms can be used to learn a policy, we consider policy gradient methods which have proven to be highly effective for dealing with large-scale problems [29, 11, 4]. These methods learn a parametrized policy $\pi_\theta(a | s)$ where θ represents the parameters; then, a gradient ascent method is employed to update parameters that would increase the expected reward of the policy in the MDP. In our work, we use a neural network to learn the policy, i.e., θ represents the weights of the network. Given a state s and action a , the policy network parametrized by θ outputs a score $H_\theta(a | s)$. Using these scores, we define the policy by the following softmax distribution: $\pi_\theta(a | s) = \frac{\exp^{H_\theta(a | s)}}{\sum_{a' \in A_s} \exp^{H_\theta(a' | s)}}$.

We use the classic REINFORCE policy gradient method (see [29, 33]) to update the weights of the network. In an episode, the RL agent performs an update as follows. First, an initial state s_0 is sampled, and then the policy π_θ is executed until the episode ends, thereby generating a sequence of experience given by $(s_t, a_t, r_t)_{t=0,1,2,\dots,L}$ where L represents the episode length. Then, in this episode, for each

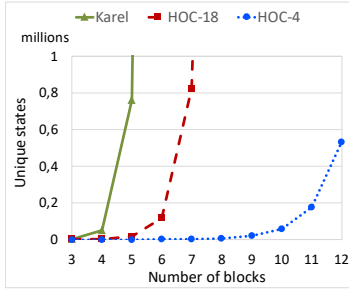
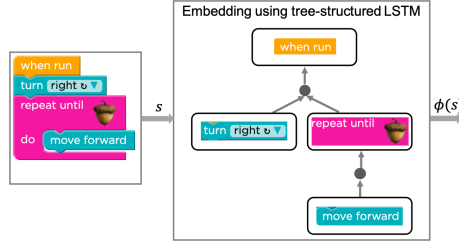
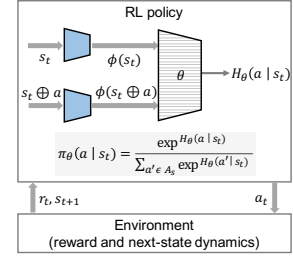


Figure 2: Number of states (i.e., unique partial solutions) grows exponentially w.r.t. the size (i.e., number of code blocks).



(a) Code embedding using tree-structured LSTM



(b) Neural architecture of our hint policy

$t \in [0, L]$, we use the following gradient update with η as learning rate:

$$\theta \leftarrow \theta + \eta \cdot \underbrace{\left(\sum_{\tau=t}^L r_\tau \right) \cdot \left(\nabla_\theta \log(\pi_\theta(a_t | s_t)) \right)}_{\text{gradient at time step } t \text{ in an episode}} \quad (2)$$

This gradient update can be computed efficiently for our setting—we refer the reader to [33, 29] for detailed discussion. We provide the implementation details in Section 4.2.

3.2 Efficient Learning of Hint Policy

3.2.1 Dealing with infinite state space

Figure 2 shows the number of states (unique partial solutions) w.r.t. the size (number of blocks) of a partial solution. Here, for reference, we also show number of states for a more complex language Karel [21]. Note that even if the correct solution is of small size (e.g., 5 for HOC-18 and HOC-4 tasks), the struggling students end up writing large partial solutions even up to 50 blocks length [23]. To deal with this computational challenge of a very large state space, we rely on code embeddings to have a featurized state representation. In our work, we train code embedding inspired by recent developments in using structured RNNs for embeddings, in particular Tree-RNN model by [22] used for HCO-18 embeddings and Tree-LSTM model by [30]. We represent the code as an Abstract Syntax Tree (AST) as shown in Figure 3a, and then this tree structure is used to process the blocks. When training, we require syntactic edit distance between raw states to be preserved after the embedding. In Section 4.2, we provide a more detailed description of the process used to learn the code embedding.

3.2.2 Dealing with state-dependant action sets

In a typical RL setting, the action set A is finite, and the standard architecture for training the network is to have $\phi(s)$ as input and the scores $H_\theta(a | s) \forall a \in A$ as output (i.e., output layer has $\mathbb{R}^{|A|}$ size). In our setting, the action set A is infinite, and the allowable actions from a state s given by set A_s are state-dependant. To tackle this challenge, we use the neural architecture as illustrated in Figure 3b. We train a network which takes as input both $\phi(s)$ and $\phi(s \oplus a)$. To evaluate the probability of taking action a from state s , we first compute scores $H_\theta(a' | s)$ for all $a' \in A_s$, and then probability of action $\pi_\theta(a | s)$ is given by the softmax distribution.

3.3 Incorporating Additional Knowledge

3.3.1 Designing rewards

We can easily incorporate additional human knowledge or features into the policy by designing appropriate reward functions [4, 11]. For instance, by changing the reward values $R(s, a)$ based on the type of action a (e.g., deleting an existing block vs. adding a new block), we can train a hint policy that favours certain types of hints. One can further incorporate more complex criterion such as suggesting hints at the last line in the code to capture students' current focus of attention which is important for better interpretability of hints [20]. Reward design also allows us to incorporate intermediate partial solutions that serve as milestones toward the final correct solution. By providing positive rewards for such states representing milestones, our hint policy would automatically learn to steer the students towards such states. Furthermore, this approach can also help in speeding-up the learning process of the RL algorithm by dealing with sparse reward problem (see Section 4.2 on how we use this idea to speed up the learning).

3.3.2 Bootstrapping from data when available

While we introduced RL framework to tackle the zero-shot challenge, the proposed framework allows one to bootstrap from additional student data if available from the same or related tasks. In fact, the existing RL-based techniques used in program synthesis and repair (see [4, 11]) have shown that substantial performance gain and convergence speed-up can be obtained by bootstrapping from available data.

We incorporate student data to bootstrap RL-based hint policy as follows. Consider we have access to a set of trajectories of successful students or experts who solved the task, given by $\Xi = \{\xi_j\}_{j=1,2,\dots}$. From these trajectories, we can obtain dataset of edits made by successful students, represented as $D_{\text{train}} = \{(s_i, s_i \oplus a_i)\}_{i=1,2,\dots}$. The RL policy network can be bootstrapped by additionally training from D_{train} using cross-entropy loss. The gradient update is given below where η' represents the learning rate:

$$\theta \leftarrow \theta + \eta' \cdot \underbrace{\left(\sum_{(s, s \oplus a) \in D_{\text{train}}} \nabla_\theta \log(\pi_\theta(a | s)) \right)}_{\text{gradient for cross-entropy loss}} \quad (3)$$

Further implementation details are provided in Section 4.2.

4. EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of our RL-based hint policy on a publicly available dataset from Code.org [5].

4.1 Hour of Code Tasks

We consider HOC-18 and HOC-4 tasks from the *Hour of Code* (HOC) challenge by Code.org, the world’s largest programming education platform [5, 23]. HOC-18 task, shown in Figure 1, is an advanced task in HOC challenge with 7 different types of blocks (“move forward”, “turn left”, “turn right”, “repeat until”, and “IF ELSE” with three different types of conditionals). HOC-4 is a simpler task with only 3 types of blocks (“move forward”, “turn left”, “turn right”). For the zero-shot setting, we do not require availability of any student data for comparing different hint-policy techniques (see Figures 4a and 4c for $x = 0$ on the x-axis). Beyond zero-shot setting, we also evaluate the performance when additional data becomes available (see Figures 4a and 4c for $x = 9, 12, 15$ on the x-axis). For these experiments, we use the trajectories of successful students from the dataset provided by [5, 23]. We refer the reader to [5, 23] for further details about these tasks and the available dataset.

4.2 Implementation Details

Here, we briefly provide implementation details for the following: (i) code embedding, (ii) RL-based hint policy, and (iii) three baselines. Some details are omitted because of lack of space—the source code would be made publicly available with the final version of the paper for reproducibility.

4.2.1 Code embedding

We learn a separate embedding for HOC-18 and HOC-4 tasks, and the code embedding is learnt prior to training the hint policy. We begin by sampling 400 random states limited to a size up to 6 blocks, and then use pairwise syntactic edit-distance between these states to generate a training dataset containing triplets of the form $(s, s', d_{\text{synt}}(s, s'))$: here $d_{\text{synt}}(s, s')$ represents the syntactic edit-distance between s and s' in terms of the number of edits required to convert s to s' . Given these triplets, we train a neural embedding $\phi(\cdot)$ so the $\|\phi(s) - \phi(s')\|_2 \approx d_{\text{synt}}(s, s')$. As shown in Figure 3a, we use the Abstract Syntax Tree (AST) representation of a state s which is then traversed in a preorder depth-first way to produce a sequence of blocks. The resulting sequence is passed through bi-directional LSTM where each unique block of the HOC language is encoded differently (cf., Tree-RNN model of [22] and Tree-LSTM model of [30]). The size of the feature representation used for our experiments is given by $\dim_\phi = 40$, i.e., $\phi(s) \in R^{40}$.

4.2.2 RL-based hint policy

For the policy network, we use a 5-layer fully connected neural network with the following architecture: (i) the input layer has $2 \times \dim_\phi$ units for $\phi(s)$ and $\phi(s \oplus a)$; (ii) the first three hidden layers have 128 hidden units and the fourth hidden layer has \dim_ϕ hidden units; and (iii) the output layer linearly aggregates \dim_ϕ values from the last hidden layer to produce the score $H(a | s)$. All hidden units use ReLU activations with a dropout rate of 0.1, and we use ADAM optimizer for training [13]. The policy actions are taken using a softmax distribution as discussed in Section 3. Below, we discuss the rewards and stopping criteria used for train-

ing, separately for zero-shot setting and when bootstrapping from available student data:

- *Zero-shot learning setting:* We set reward $R(s, a)$ as +100 when $s \oplus a = s^*$, and -1 otherwise. The training is done until the average reward of the policy is saturated. To further speed up the convergence, we use intermediate rewards in the training process by adding an additional term of $-d_{\text{synt}}(s \oplus a, s^*)$ to the reward. Here, d_{synt} represents the syntactic edit-distance between two states (same function as used in generating training data for code embedding). These intermediate rewards during the training process allowed us to speed up the convergence by order of magnitude, without effecting the overall performance of the trained policy. After this speed up, the number of episodes required until convergence varied from 5000 to 20,000.
- *Additional student data is available:* We first pre-train the network using cross-entropy loss with the data sampled from D_{train} . This pre-training is done for 20 epochs where each epoch consists of multiple gradient updates as follows: A batch of data is sampled from D_{train} of size given by $\text{batchsz} = 32$ and a gradient update is performed using this batch as per Eq. 3; this process is repeated $\frac{|D_{\text{train}}|}{\text{batchsz}}$ within an epoch. After this pre-training phase, we train the policy network using rewards for 2000 episodes using the gradient updates in Eq. 2. Given that the pre-training phase already provides a good initialization of the policy network, we used modified reward signals in this case as compared to the zero-shot setting: (i) we set reward of +20 for reaching the goal instead of +100 and (ii) we reduced the value of intermediate rewards and set it to $-0.05 \cdot d_{\text{synt}}(s \oplus a, s^*)$ by scaling it down.

4.2.3 Baselines

We compare our RL-based hint policy REINFORCE-HP w.r.t. several baselines as discussed below. In particular, we consider baseline techniques which can be implemented efficiently, without requiring any explicit graph representation of the state space which is computationally intractable (also, see Figure 2).

As a simple benchmark, we use RANDOM-HP: a baseline policy that simply selects an action $a \in A_s$ randomly when providing a hint for state s . As another natural baseline, we consider FREQNEXT-HP which uses historical student data as follows. Based on the available data, a frequency count $\text{count}(s, a)$ is maintained for each (s, a) pair counting the number of times action a was taken from state s by students in the historical data. Then, when providing hint for a state s , the hint is chosen from a distribution given by the following softmax distribution: $P(a|s) = \frac{\exp(1 + \text{count}(s, a))}{\sum_{a' \in A_s} \exp(1 + \text{count}(s, a'))}$.

Next we discuss a baseline based on the state of the art technique of *Continuous Hint Factory* (CHF) [20] that uses supervised learning approach. We adapt the key ideas of CHF to our setting and refer to the resulting hint policy as REGRESSION-HP—this adaption allows us to directly compare REINFORCE-HP with REGRESSION-HP as both these hint policies use the same embedding and same historical data when available. Below, we discuss three key steps required in training REGRESSION-HP:

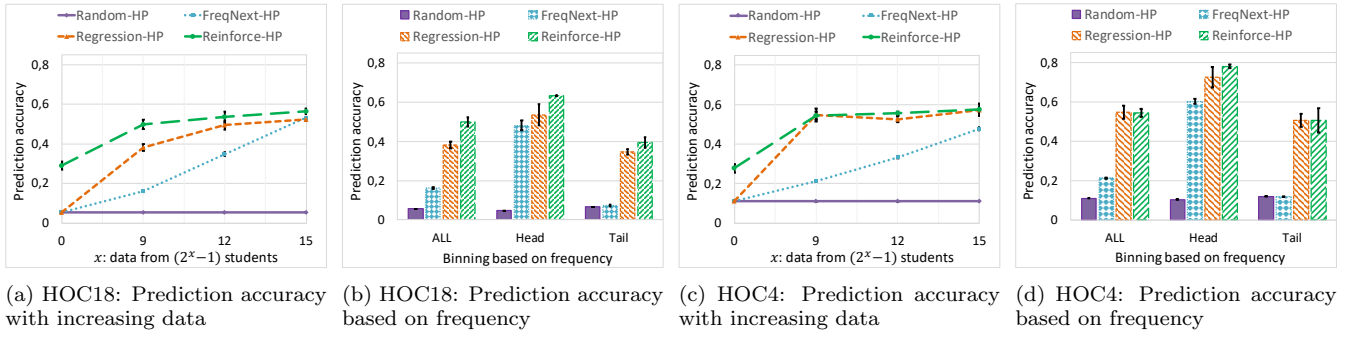


Figure 4: **(a)** shows prediction accuracy for HOC-18 task. $x = 0$ on the x-axis corresponds to the zero-shot setting. REINFORCE-HP achieves over 20% absolute improvement in the prediction accuracy compared to baselines. **(b)** shows results for HOC-18 task when states are binned into “Head” and “Tail” based on frequency counts, and a moderate amount of historical student data is available (see details in Section 4.3). REINFORCE-HP performance on low-frequency states is even higher than the overall performance of any of the baselines. **(c, d)** shows the results for simpler task of HOC-4.

- *Embedding* (cf. Section 3.1 of [20]): For REGRESSION-HP, we use the same code embedding as used for REINFORCE-HP. Our code embedding is similar to the Euclidean embedding space used by [20] which was obtained by preserving syntactic distances between raw states.
- *Regression function* (cf. Section 3.2 of [20]): Then, we learn a regression function in the embedding space which can identify the most likely hint as a vector in this space. This step makes use of available student data D_{train} as discussed in Section 3.3, and learns a function f_{reg} that can map $\phi(s)$ to $\phi(s \oplus a)$ for $(s, s \oplus a) \in D_{\text{train}}$. We use neural network to learn this function f_{reg} , in contrast, [20] used Gaussian process regression. We use a 4-layer neural network to learn f_{reg} with essentially the same architecture as the one used to learn REINFORCE-HP, except that (i) the input layer has \dim_{ϕ} units for $\phi(s)$, (ii) the output layer here has \dim_{ϕ} units to produce $\phi(s \oplus a)$ (this corresponds to what was the last hidden layer in REINFORCE-HP neural architecture).
- *Human-readable hint* (cf. Section 3.3 of [20]): Finally, when providing hint for a state s , we first compute the hint in embedding space as $f_{\text{reg}}(\phi(s))$ and then convert this to an editable hint $a \in A_s$ as the one that minimizes $\|f_{\text{reg}}(\phi(s)) - \phi(s \oplus a)\|_2$.

4.3 Results

Figure 4 shows the results for HOC-18 and HOC-4 tasks, averaged over 3 runs of all the hint policies. Figures 4a and 4c show the overall average prediction accuracy. The $x = 0$ point in these plots corresponds to the zero-shot setting and measures the prediction accuracy of next-step hint for the very first student who is attempting the task. For both HOC-18 and HOC-4 tasks, our RL-based policy REINFORCE-HP has a significant improvement over baselines by about 20% gain in absolute accuracy. For the HOC-18 task, even when a moderate amount of data becomes available (e.g., see $x = 9$ on the plot which is equivalent to data of 511 students), REINFORCE-HP improves w.r.t. REGRESSION-HP by 10% gain in absolute accuracy.

In Figures 4b and 4d, we further analyze the performance of different hint policies when training using a moderate amount of available data (corresponding to $x = 9$ in Fig-

ures 4a and 4c which is equivalent to data of 511 students). In these plots, states are binned into “Head” and “Tail” based on frequency counts as available in the dataset obtained from [23]. Here, the bin “Head” corresponds to top 40 states and “Tail” corresponds to bottom 40 states based on frequency. For HOC-18 task, REINFORCE-HP performance on low frequency states is even higher than the overall performance of any of the baselines: this highlights the power of RL framework that allows an efficient self-exploration of the solution space when learning the hint policy. These plots also illustrate that techniques such as FREQNEXT-HP that rely on frequency counts can have much worse performance on the tail segment of states compared to head segment of states.

In summary, these results demonstrate that the proposed RL framework enables us to learn an effective hint policy in the zero-shot setting, and the performance can be further improved with the availability of student data.

5. CONCLUSIONS AND FUTURE WORK

We tackled the challenge of zero-shot learning of hint policy to be able to provide hints for the very first student working on a coding task. Building on the recent advances in RL-based neural program synthesis, we proposed an RL framework for learning hint policy. Using a publicly available dataset from Code.org, we showed that our policy achieves significant improvements over state of the art supervised learning techniques when no or very limited data is available. Furthermore, the results demonstrated that our proposed framework is easily amendable, e.g., it can benefit from historical student data to further boost the performance.

There are several research directions for future work. As an evaluation criterion, we used the prediction accuracy of next-step hints based on expert annotations. In future work, it would be important to do user studies and understand the pedagogical value of these hints. In this work, our hint policy provided hints based on only the current partial solution of the student. It would be important to learn a richer hint policy that can provide personalized hints by accounting for the whole trajectory of the student. Finally, it would be interesting to apply our framework to more complex learning scenarios (e.g., with more complex coding tasks or with a more complex language involving additional concepts such as the ability to declare variables).

6. REFERENCES

- [1] U. Z. Ahmed, S. Gulwani, and A. Karkare. Automatically generating problems and solutions for natural deduction. In *IJCAI*, 2013.
- [2] V. Aleven, I. Roll, B. M. McLaren, and K. R. Koedinger. Help helps, but only so much: Research on help seeking with intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, 26(1):205–223, 2016.
- [3] T. Barnes and J. Stamper. Toward automatic hint generation for logic proof tutoring using historical student data. In *International conference on intelligent tutoring systems*, pages 373–382. Springer, 2008.
- [4] R. Bunel, M. J. Hausknecht, J. Devlin, R. Singh, and P. Kohli. Leveraging grammar and reinforcement learning for neural program synthesis. In *ICLR*, 2018.
- [5] Code.org. Code.org: Learn computer science. <https://code.org/research>.
- [6] J. Devlin, R. Bunel, R. Singh, M. J. Hausknecht, and P. Kohli. Neural program meta-induction. In *NIPS*, pages 2080–2088, 2017.
- [7] A. Ghosh, S. Tschischek, H. Mahdavi, and A. Singla. Towards deployment of robust cooperative ai agents: An algorithmic framework for learning adaptive policies. In *AAMAS*, 2020.
- [8] S. Gross, B. Mokbel, B. Paassen, B. Hammer, and N. Pinkwart. Example-based feedback provision using structured solution spaces. *International Journal of Learning Technology* 10, 9(3):248–280, 2014.
- [9] S. Gross and N. Pinkwart. How do learners behave in help-seeking when given a choice? In *International Conference on Artificial Intelligence in Education*, pages 600–603. Springer, 2015.
- [10] S. Gulwani, O. Polozov, R. Singh, et al. Program synthesis. *Foundations and Trends® in Programming Languages*, 4(1-2):1–119, 2017.
- [11] R. Gupta, A. Kanade, and S. K. Shevade. Deep reinforcement learning for syntactic error repair in student programs. In *AAAI*, pages 930–937, 2019.
- [12] B. Kartal, N. Sohre, and S. J. Guy. Data driven sokoban puzzle generation with monte carlo tree search. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2016.
- [13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [14] J. Krajcik. Three-dimensional instruction: Using a new type of teaching in the science classroom. *Science Scope*, 39(3):16, 2015.
- [15] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350:1332–1338, 2015.
- [16] T. Lazar and I. Bratko. Data-driven program synthesis for hint generation in programming tutors. In *International Conference on Intelligent Tutoring Systems*, pages 306–311. Springer, 2014.
- [17] Z. Manna and R. J. Waldinger. Toward automatic program synthesis. *Communications of the ACM*, 14(3):151–165, 1971.
- [18] J. Oh, S. Singh, H. Lee, and P. Kohli. Zero-shot task generalization with multi-task deep reinforcement learning. In *ICML*, pages 2661–2670, 2017.
- [19] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.
- [20] B. Paaßen, B. Hammer, T. W. Price, T. Barnes, S. Gross, and N. Pinkwart. The continuous hint factory - providing hints in continuous and infinite spaces. *Journal of Educational Data Mining*, 2018.
- [21] R. E. Pattis. *Karel the robot: a gentle introduction to the art of programming*. John Wiley & Sons, Inc., 1981.
- [22] C. Piech, J. Huang, A. Nguyen, M. Phulsuksombati, M. Sahami, and L. J. Guibas. Learning program embeddings to propagate feedback on student code. In *ICML*, pages 1093–1102, 2015.
- [23] C. Piech, M. Sahami, J. Huang, and L. J. Guibas. Autonomously generating hints by inferring problem solving policies. In *Conference on Learning @ Scale, L@S*, pages 195–204, 2015.
- [24] T. W. Price, Y. Dong, and D. Lipovac. isnap: towards intelligent tutoring in novice programming environments. In *SIGCSE*, pages 483–488, 2017.
- [25] B. Priemer, K. Eilerts, A. Filler, N. Pinkwart, B. Rösken-Winter, R. Tiemann, and A. U. Zu Belzen. A framework to foster problem-solving in stem and computing education. *Research in Science & Technological Education*, pages 1–26, 2019.
- [26] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1st edition, 1994.
- [27] K. Rivers and K. R. Koedinger. Data-driven hint generation in vast solution spaces: a self-improving python programming tutor. *International Journal of Artificial Intelligence in Education*, 27(1):37–64, 2017.
- [28] R. Singh, S. Gulwani, and S. Rajamani. Automatically generating algebra problems. In *AAAI*, 2012.
- [29] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [30] K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, pages 1556–1566, 2015.
- [31] K. Vanlehn. The behavior of tutoring systems. *International journal of artificial intelligence in education*, 16(3):227–265, 2006.
- [32] V. K. Verma, G. Arora, A. Mishra, and P. Rai. Generalized zero-shot learning via synthesized examples. In *CVPR*, pages 4281–4289, 2018.
- [33] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.
- [34] M. Wu, M. Mosse, N. Goodman, and C. Piech. Zero shot learning for code education: Rubric sampling with deep learning inference. In *AAAI*, volume 33, pages 782–790, 2019.
- [35] K. Zimmerman and C. R. Rupakheti. An automated framework for recommending program elements to novices (n). In *International Conference on Automated Software Engineering, ASE*, pages 283–288, 2015.

Investigating Relations between Self-Regulated Reading Behaviors and Science Question Difficulty

Effat Farhana
North Carolina State University
efarhan@ncsu.edu

Teomara Rutherford
University of Delaware
teomara@udel.edu

Collin F. Lynch
North Carolina State University
cflynch@ncsu.edu

ABSTRACT

Reading to learn is a quintessentially self-regulated activity. In order to provide effective support for this activity it is necessary for us to understand how students adapt their self-regulation behaviors within disciplinary reading environments. In this paper, we utilize student response data from a digital literacy platform to examine the association of students' behaviors with the difficulty of questions embedded in science texts. We analyzed 131 distinct physical science questions used in 641 middle school classes within *Actively Learn*, a digital reading platform. We investigated the association of question difficulty and students' behaviors, including reading, annotating, highlighting, and vocabulary lookups. Our findings show that students found multiple choice questions with multiple correct answers hard to answer and exhibited more reading behaviors when attempting them. Short answer questions appeared to be easier; students engaged in more annotation, highlighting vocabulary lookups when attempting easy short-answer questions compared to difficult multiple-choice questions.

Keywords

Question Difficulty, Student Behavior, Self-Regulated Learning

1. INTRODUCTION

Reading to learn, as students do when engaging with disciplinary texts [35], is a quintessentially self-regulated activity [26]. When presented with a block of text, students can approach it by reading end to end, make notes as they go or not. They can also skip around for clues, or even explore in larger chunks. How they choose to do so will be driven by their own study habits [38], as well as the context of the assignment itself.

Students who are trying to answer a set of questions typically read differently than students who are trying to master general material [11,18]. As the questions change, so will their behavior. They will, to paraphrase Karl Llewellyn, *read with new eyes* [23]. In order to effectively support students in reading to learn, it is necessary to understand how students adapt their reading and learning strategies when faced with problems at different perceived levels of difficulty and of different types. Understanding these changes will allow us to model their behaviors, identify successful and unsuccessful approaches, and provide effective interventions as necessary.

Prior researchers have shown that reading scientific texts requires both reading strategies and self-regulated learning (SRL) strategies [14, 25, 47]. As Butler and Cartier emphasized, understanding SRL requires understanding students' learning contexts [9]. The context of learning is nested: geographical, socio-economical, within-school, and within-classroom. At the classroom level, students' engagement in learning is shaped by teacher's instructional approaches and by interactions with the teacher and peers [9].

Our goal in this study is to examine how students may perceive question difficulty at the *class-level*, and how students vary their individual reading and self-regulated learning activities in response to it. The context of our study is *Actively Learn* (AL) [1], an online reading platform that is used in schools in the United States. For this study, we focus on readings and test items in middle school science domains. We answer the following research questions:

RQ 1. How does students' performance vary with question difficulty?

RQ 2. What SRL strategies do students use before and after each question ?

2a. How did SRL strategies vary with question difficulty?

2b. How did reading vary with question difficulty?

We collected log data from 11,832 middle school physical science students within the AL platform. We extracted reading, annotating, highlighting, and vocabulary lookup events from the log traces and we estimated the difficulty level of questions by class level. We compared our difficulty level with a comparable analysis from item response theory (IRT) [18]. And we

Effat Farhana, Teomara Rutherford and Collin Lynch
"Investigating Relations between Self-Regulated Reading Behaviors and Science Question Difficulty" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 395 - 402

evaluated students' reading and SRL strategy usage with question difficulty.

2. LITERATURE REVIEW

Our research draws on prior work in two primary areas: research on self-regulated learning activities in reading-to-learn situations and research on question difficulty analysis from student performance data.

2.1 SRL and Science Achievement

SRL, as described by Zimmerman, involves four regulatory components during learning: goal setting, self-monitoring, self-evaluating, and using strategies to control progress toward a goal [51]. Learners who are more capable at self-regulation tend to set more challenging goals for their academic achievement than those who are less capable [53]. They use self-monitoring strategies to monitor their time on task and to solve conceptual problems [8]. Self-evaluation, in this context, means being able to judge the outcomes of self-monitoring processes [52]. In the process of self-evaluation, a student changes learning strategies to achieve their learning goals [53]. Prior researchers have provided a range of SRL models, these include Pintrich's SRL framework [31], Zimmerman's cyclic phases model [50] and Winne and Hadwin's model [46]. While they rely on different assumptions, all of them frame learning as an active process wherein learners set goals by understanding topics or domains, regulate their cognition processes, and modify behaviors to achieve goals in light of self-evaluation [47, 31].

SRL strategies are linked to subject domains [48]. Researchers have examined SRL strategy usage and academic performance in science in game-based learning [37, 40], classroom settings [4], and in agent-based learning environments [7]. Francois et al. examined students' SRL usage strategies in an agent-based learning environment for human biology, MetaTutor [7]. They found high performing students both took more notes and made more summaries. Low performing students, by contrast, struggled to find relevant pages to attain their subgoals within the system. Andrzejewski et al. examined an SRL intervention in a 9th grade earth science class [4]. They found SRL intervention strategies had different effects on students with different socioeconomic status. Students from minority groups (non-white or economically disadvantaged) benefited more than those in the majority group (white and middle class). Rutherford examined the role of SRL within a curriculum integrated mathematics game, ST Math, and found that differences in students' SRL monitoring was related to their academic performance [37].

Our goal in this analysis is to evaluate students' SRL usage in middle school science reading. Our work is situated in the interactions between SRL monitoring and control—as students engage with text and with embedded questions, they assess the difficulty of the task they encounter and adjust their behaviors accordingly. We operationalize the SRL activities related to reading strategies students would use during the control phase of SRL as annotating [24], highlighting [45], and vocabulary lookups, as we believe that these features serve as proxies for SRL behaviors, and we have studied their relation to question type in a prior publication [16]. Science texts involve key concept words and vocabulary terms. Students' reading comprehension and motivation has been found to decrease due to introduction of concept words [22]. Vocabulary lookups can help students to understand concepts when they first encounter

them. Annotation requires that students comprehend text and frame it in their own words [24]. Highlighting texts involve SRL activities through the use of monitoring information and connecting that information to prior knowledge [45].

2.2 Question Difficulty from Student Data

Understanding the difficulty level of test items has a wide range of applications in educational data mining (EDM); this includes work on the optimal arrangement of curricula [21] and on the design of adaptive tests or personalized learning environments and intelligent tutoring systems (ITS) [30]. Item difficulty can be assessed based upon the design of a question and its classroom context [20], or it can be evaluated empirically based on observed student performance in real contexts [28]. This empirical approach is particularly important for the development of practical adaptive learning and tutorial environments. Although the structure of a question specifies the knowledge required, the *operational difficulty* of a task, that is the difficulty for a given student, is dependent upon the class context, the amount of individual preparation or scaffolding provided, the students' skill level, and whether they are working on it individually, as part of a team, or as a whole class.

Consequently, a number of prior EDM researchers have developed a number of domain and student models which can be used to identify structural relationships between tasks and to assess their difficulty based upon empirical performance. These efforts include: work on q-matrices that map items to required skills and levels (e.g., [5]); learning factors analysis and other student performance models such as Bayesian Knowledge Tracing (BKT) (e.g., [10, 13]); and item response theory (IRT) [19]. Item Response Theory (IRT) is regarded as the “gold standard” of estimating question difficulties from student response data. The simplest version of IRT is the “Rasch Model” [32], which associates a skill or ability to each student and a difficulty level to each question.

Different intelligent tutoring systems (ITS) [44] and other learning environments have utilized student-system interaction logs to estimate question difficulty empirically. Pardos and Heffernan for example, extended the BKT model to handle item difficulty in a mathematics tutoring system, ASSISTment [30]. QuizGuide, an assessment system for Java programming [39], predicts subjective difficulty on questions from predefined weights and student performance. The predefined weights were assigned by domain experts. ELM-ART II, a web based Lisp programming tool [40], uses fixed difficulty and weight for each item. A student's knowledge level is updated based on correct or incorrect attempts on each item and difficulty level. Researchers have further utilized student attempts coupled with IRT to estimate question difficulty [33]. Fouh et al., for example, utilized the total number of attempts and guessing behavior to understand difficult topics in a Data Structure course [17]. Additionally, they compared their approach to IRT.

As in this prior work, we focus on using student-system interaction logs to estimate the operational difficulty of our questions; however, as we are particularly interested in variation across instructors, we analyze our data at the class level.

3. DATASET

In this section we describe the Actively Learn platform [1] and our dataset construction process.

3.1 The Actively Learn (AL) Platform

AL is a digital literacy platform aimed at students in primary and secondary (K-12) education. AL is designed to improve students' reading proficiency. The platform allows teachers to assign reading texts as assignments to class with embedded questions, which may include optional automated feedback. Assignments in the AL platform can range from one page to multiple pages. Questions in AL can be multiple choice (MCQ) and short answer (SA), including free texts and fill in the blanks. Teachers may use predefined reading texts and questions available within AL or introduce their own as assignments. MCQs are automatically graded, whereas SAs are not. AL questions are graded on a scale of zero to four. Figure 1 shows a reading text in the AL interface.

Physical science reading texts in the AL platform are organized following the Next Generation Science Standards (NGSS) guidelines [2]. The NGSS for middle school physical science (PS) has four standards: (i) PS1: Matter and its Interactions, (ii) PS2: Motion and Stability: Forces and Interactions, (iii) PS3: Energy, and (iv) PS4: Waves and their Applications in Technologies for Information Transfer. Students are expected to analyze and interpret data (PS1 standard), plan and carry out investigations (PS2 standard), develop and use models, analyze data (PS3 standard), and use mathematical thinking and demonstrate understanding (PS4 standard) [2].

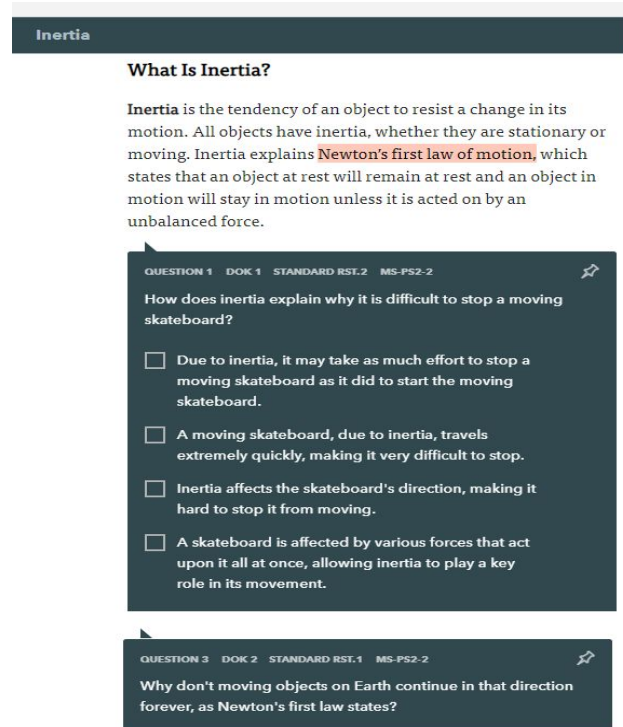


Figure 1. A reading text and embedded questions. Question 1 is an MCQ and question 3 is a SA.

AL's developers state that the platform provides opportunities for teachers and students to deeply engage with text [34].

Students can highlight, annotate, and look up unknown words as they proceed through the readings.

3.2 Dataset Construction

Our current study focuses on middle school science reading assignments in the AL platform. Our dataset includes records of students who completed assignments in 2018. Our dataset includes 17,886 student records across 1,033 classes. After plotting histograms of class sizes, we excluded classes with fewer than 10 or more than 60 students. This left us with 83.45% of students. We also excluded any student enrolled in multiple classes, as we believed these accounts could be for testing purposes. After selecting classes, we filtered the dataset by questions. We selected 131 predefined AL questions used in at least two classes. The final resulting dataset has 11,832 students and 913 assignments used in 641 classes. We extracted students reading, highlighting, annotating, and vocabulary lookup events from log data trace.

4. METHODOLOGY and RESULTS

In this section we describe our methodology to answer our RQs.

4.1 RQ1: How does students' performance vary with question difficulty?

In our study, a question can be used by different classes. As we do not have access to student demographics and other confounding variables, we opted to aggregate difficulty data at the level of classes. Additionally, we compared our approach with the IRT model. Note that estimating question difficulty is not the goal of our study. We aimed to investigate how students' reading and SRL strategy usage varies with question difficulty. In order to analyze how students respond to different questions, it is necessary to identify suitable metrics to assess question difficulty. First we defined metrics to assess each question difficulty within a class from student interaction data. We analyzed how a question's perceived difficulty varies across classes using our defined metrics. We assessed students' performance on questions categorized by question difficulty. Next, we performed IRT analysis to examine the relationship between question difficulty and student performance. We compared findings between two approaches.

4.1.1 Question Difficulty and Student Performance: Student Interaction Data

We analyzed the students' performance on each question to assess the difficulty of the question. To calculate a student's performance, we took the ratio of max score achieved to number of attempts on a question. Questions in AL are graded on a scale [0-4]. For our assessment, we normalized the students' scores to a range of [0-1]. We defined the performance of a student i on a question q as

$$r_i = \frac{\text{scaled maximum score on } q}{\text{no. of attempts on } q} = \frac{\text{maximum score on } q/4}{\text{no. of attempts on } q} \quad (1)$$

Equation (1) computes a student's score of a question on a scale of zero to one, one representing good performance and zero representing poor performance.

We computed difficulty level (dl) of a question q as

$$dl = 1 - \frac{\sum_{i=1}^n r_i}{n} \quad (2)$$

where n is the number of students in a class who attempted q , and r is the students' performance on q as defined above. A $dl \sim 0$ value indicates an easy question and $dl \sim 1$ indicates a difficult one.

To analyze the difficulty of a question q across classes, we computed difficulty ratio of q across classes as follows:

$$\text{Difficulty ratio of question } q = \frac{\text{No. of classes with } dl \geq 0.5 \text{ for } q}{\text{No. of classes used } q \text{ in assignments}} \quad (3)$$

We plotted histograms of difficulty ratio for 131 questions. After examining the histograms, we observed more questions with difficulty ratio < 0.2 and fewer questions with difficulty ratio > 0.5 . We grouped questions into three categories by their difficulty ratio as shown in Table 1.

We plotted histograms of student performance on each question, r , for three categories of questions. Figure 2 presents the histograms (next page).

4.1.2 Question Difficulty and Student Performance: IRT Analysis

The IRT method estimates the probability of a student getting an item correct based upon the item difficulty and the students' ability. We applied the 1-parameter logistic IRT model (1PL) model, also known as the Rasch model. The 1PL model describes test items considering only one parameter, *item difficulty*, b . The 1PL model is a logistic curve, i.e., it evaluates how high the latent ability level needs to be in order to get a 50% chance of getting the item right. Item difficulty is estimated from the student responses.

Table 1: Question category by difficulty ratio (diff. ratio)

Question Category	MCQ	SA	Total
<i>Easy</i> (diff. ratio < 0.4)	6	75	81
<i>Medium</i> ($0.4 \leq \text{diff. ratio} \leq 0.6$)	5	26	31
<i>Hard</i> (diff. ratio > 0.6)	11	8	19

The Rasch model assumes a boolean score for each student response to questions. To apply the 1PL model, we need to map students' responses to 0 or 1 computed from equation (1). We assigned zero if $r < 0.5$ and 1 otherwise. We fit the 1PL model to 131 questions using the 'ltm' package in R [36].

We plotted per-item characteristic curves (ICC) from the fitted model. The X axis of the ICC represents students' latent ability and the Y axis represents the probability of answering the question correctly. The range of the X axis is $[-4, 4]$, where zero indicates *average* ability. We plotted ICC curves for *Easy*, *Medium*, and *Hard* questions separately. We also plotted item information curves (IIC) from the fitted model. The IIC curves shows how much information about students' ability an item provides. A difficult item will provide little information about a

student with low ability and vice versa for easy items. We plotted IIC curves for *Easy*, *Medium*, and *Hard* questions separately.

4.1.3 Results for RQ1

From the student interaction results shown in Figure 2. We also notice the number of students receiving zero in *Easy* questions is higher than *Medium* and *Hard* ones.

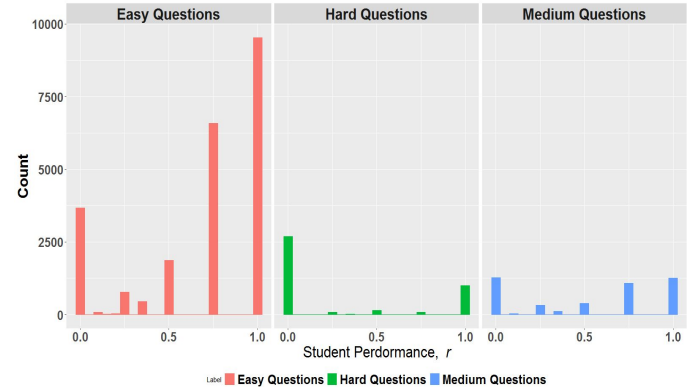


Figure 2. Student performance by question difficulty

In Figure 3 (next page) we show our ICC results for *Easy*, *Medium*, and *Hard* questions. Each line represents the ICC curve of one question. We observe that the ICC curves for *Easy* questions are mostly on the left side of zero, indicating *Easy* questions required lower ability for correct attempts. Comparing ICC curves of *Easy* and *Hard* questions, we note that *Hard* questions have curves more on the right side of the X axis. The probability of answering a *Hard* question correctly decreases as curves go from left to right.

The IIC curve shows how much information about students' ability a question gives. From Figure 3, we observe *Easy* questions curves provide information about students with average and below average abilities (the peak of curves are mostly on the left side of $X = 0$. $X = 0$ refers to average ability). Similarly, IIC curves for *Hard* questions provide information about high ability (the peak of curves are mostly on the right side of $X = 0$) levels.

4.2 What SRL Strategies Do Students Use Before and After Questions?

In this section we present our methodology and results for RQ2. We calculated SRLs at student-level to understand how students' SRLs varied by question difficulty.

4.2.1 Methodology for RQ2

To investigate the association between students' reading and SRL behavior with question difficulty, first we need to identify student sessions. The AL system does not record student sessions. Therefore, we relied on a data-driven approach to identify sessions as described by Kovanovic et al. [41] and Adithya et al. [3]. AL records timestamps of students' question submission, reading, annotating, highlighting, and vocabulary lookup behaviors. We aggregated timestamps of students' actions into a unified log. We plotted histograms of time intervals between consecutive actions to identify outliers and estimate the last action of any time period [41].

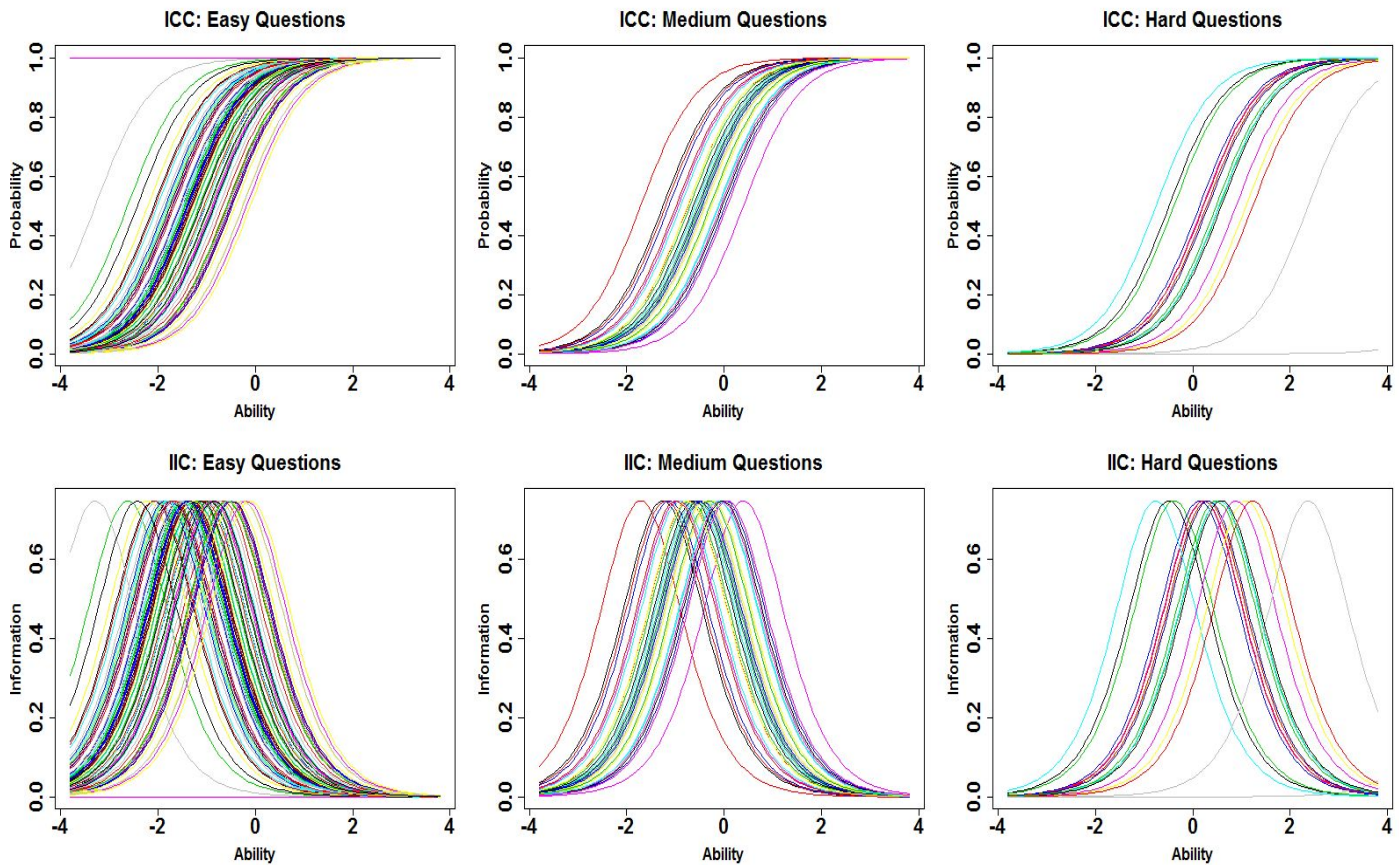


Figure 3. ICC and IIC plots from 1PL model

After conducting this analysis, we selected 30 minutes as a session. Any time interval greater than 30 minutes was marked as the beginning of a new session.

We then split students' actions into sessions. Next, we counted reading and SRL activities prior and after each question submission. We calculated the mean and standard deviation for the four reading and SRL features. To test if there were statistically significant differences in means, we applied the nonparametric Kruskal-Wallis test. In cases with statistically significant differences in mean, we performed a post-hoc Dunn test with Benjamini-Hochberg correction to identify pairwise statistically significant groups, using the R package "dunn.test" [15]. Table 2 presents the mean, standard deviation, and p value from Kruskal-Wallis test.

4.2.2 Results for RQ2

In this section we present our results to answer RQ2 and the sub-questions:

2a. How did SRL strategies vary with question difficulty?

2b. How did reading vary with question difficulty?

As Table 2 shows, the mean of all features vary at statistically significant levels across the three categories of questions. Number of reading activities is the highest for the *Hard* questions, followed by *Medium*, and *Easy*. This indicates students had to read more prior to attempting a *Hard* question.

Table 2: Mean with (Standard Deviation), and p value from KW = Kruskal-Wallis test for student behavior features on *Easy*, *Medium*, and *Hard* questions. R = Reading, A = Annotating, H = Highlighting, V = Vocabulary lookups

Feature	<i>Easy</i>	<i>Medium</i>	<i>Hard</i>	KW p
R	0.684 (0.71)	0.814 (0.74)	1.27 (0.70)	< 0.001
A	0.335 (0.20)	0.021 (0.17)	0.012 (0.12)	< 0.001
H	0.007 (0.10)	0.004 (0.06)	0.002 (0.05)	< 0.001
V	0.015 (0.13)	0.014 (0.12)	0.009 (0.1)	0.01

It also indicates that they revisited the reading material after attempting a *Hard* question more frequently than they did for *Easy* and *Medium* questions. Annotating, highlighting, and vocabulary lookup counts were higher in *Easy* and *Medium* questions as compared to *Hard* ones. We report the Dunn test and statistically significant pairs for each feature below. We report effect-size (r) using a nonparametric test, Cliff's-Delta [12].

For the reading feature (R), we found statistically significant differences among all three pairs *Easy-Hard*, *Easy-Medium*, and *Medium-Hard*. The p values of these pairs were *Easy-Hard* ($p < 0.001$, $r = 0.43$), *Easy-Medium* ($p < 0.001$, $r = 0.10$), *Medium-Hard* ($p < 0.001$, $r = 0.34$)

When we consider the annotating feature (A), we also found statistically significant differences in means among all three pairs. *Easy-Hard*, *Easy-Medium* and *Hard-Medium* pairs had ($p < 0.001$, $r = 0.02$), ($p < 0.001$, $r = 0.012$), and ($p = 0.018$, $r = 0.01$), respectively.

And, when considering the highlighting feature (H), we found two pairs differed at statistically significant levels: *Easy -Hard* ($p = 0.004$, $r = 0.004$) and *Easy-Medium* ($p = 0.0209$, $r = 0.003$).

Finally, for the vocabulary lookup (V) feature, we found one pair with a statistically significant difference: *Easy-Hard* ($p = 0.005$, $r = 0.01$).

5. DISCUSSION

We summarize our findings and implications of results below.

In this study we used a data-driven approach on class-level student response data to group questions by difficulty levels. Our difficulty levels are consistent with findings from IRT analysis. ICC curves for *Easy* questions require lower student ability (Figure 3) and vice versa for *Hard* questions.

Table 1 shows 11 MCQ questions belonging to the *Hard* category. We looked into the question texts and observed 10 out of 11 questions required students selecting multiple options, e.g., "Select all that apply." Our analysis from RQ2 indicates students exhibited more reading (R) behavior prior and after answering *Hard* questions compared to *Easy* and *Medium* ones. Thus, our findings indicate that although students can often rule out distractors in MCQs [6], answering such questions is *Hard* when options involve selecting multiple correct answers. Our findings may be helpful for ITS designers. Developers of ITS can facilitate more hints on MCQ questions having multiple correct answers, so that students do not find those *Hard*.

From Table 1, we observed 75 out of 81 *Easy* questions were SAs. Our results for RQ2 indicated that students annotated (A), highlighted (H), and looked up vocabulary (V) more in answering *Easy* questions. We conclude that the format of questions may have contributed to students' SRL usage, even if the difficulty level was classified as *Easy*. Ideally, we would have been able to control question format and student characteristics; secondary data mining allows for large-scale data, but precision of results can be compromised by lack of these details. Nevertheless, we were able to demonstrate that SRL behaviors covary with question difficulty and/or format. It seems likely that as students encountered SA questions, they received metacognitive signals that encouraged their use of SRL behaviors [27] and this resulted in the relatively greater success of these questions. However, we cannot disentangle this from difficulty in our data. Although multiple option MCQs were difficult for students, they may not have triggered metacognitive awareness of the need for SRL behaviors. This is in line with some prior research suggesting less confidence bias in SA questions than in MCQs [29].

6. LIMITATIONS

Our study has two limitations. First, student responses to assignment questions are dependent on the teacher's selection of questions. We do not have responses to all questions for every student. Thus, the latent ability analysis of IRT is limited to student response data. Second, we did not consider the text complexity of the reading article in analyzing question difficulty. Science reading requires analyzing information from texts, diagrams, mathematical equations, and videos [22, 49]. Future research direction can investigate the association of question texts and the reading texts to understand text complexity.

7. CONCLUSION

In this study we investigated associations of students' reading and SRL behavior with question difficulty in middle school science reading. We analyzed question difficulty at the class level and compared our analysis method with IRT. Our results show that MCQ with multiple correct options are generally harder for students in our middle-school set. And we show that when faced with such hard questions, irrespective of their type, students engage in more reading activities but not the other SRL actions we measured. *Easy questions, by contrast, were more commonly* SAs than MCQs. Students spent more time annotating, highlighting, and looking up vocabulary terms in *Easy* questions. This may reflect that the easy questions in our dataset are more focused on rote memorization or on localizing responsive passages in the larger text than on concept synthesis or summarization, or, alternately, SA questions may prompt students to engage in SRL behaviors that MCQs do not. Due to the confounding of difficulty and format type, we were unable to disentangle these reasons. We hope our work opens up further opportunities for researchers and ITS developers to explore student interaction with question difficulty.

8. REFERENCES

- [1] Actively Learn. <https://www.activelylearn.com/>
- [2] Next Generation Science Standards. <https://www.nextgenscience.org/>.
- [3] S. Adithya, N. Gitinabard, C. F. Lynch, T. Barnes, and S. Heckman. Predicting student performance based on online study habits: A study of blended courses. In *International Conference on Educational Data Mining*, 2018.
- [4] C. E. Andrzejewski, H. A. Davis, P. S. Bruening, and R.R. Poirier. Can a self-regulated strategy intervention close the achievement gap? exploring a classroom-based intervention in 9th grade earth science. *Learning and Individual Differences*, 49:85-99, 2016.
- [5] T. Barnes. The q-matrix method: Mining student response data for knowledge. In *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, 2005.
- [6] L. B. Bliss. A test of lord's assumption regarding examinee guessing behavior on multiple-choice tests using elementary school students. *Journal of Educational Measurement*, pages 147 -153, 1980.
- [7] F. Bouchet, R. Azevedo, J. S. Kinnebrew, and G. Biswas. Identifying students' characteristic learning

- behaviors in an intelligent tutoring system fostering self-regulated learning. *International Educational Data Mining Society*, 2012.
- [8] T. Bouffard-Bouchard, S. Parent, and S. Larivee. Influence of self-efficacy on self-regulation and performance among junior and senior high-school age students. *International Journal of Behavioral Development*, 14(2):153-164, 1991.
- [9] D. L. Butler and S. C. Cartier. 2005. Multiple Complementary Methods for Understanding Self-Regulated Learning as Situated in Context. In *Annual Meetings of the American Educational Research Association, Montreal, QC (2005)*
- [10] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis-a general method for cognitive model evaluation and improvement. In *International Conference on Intelligent Tutoring Systems*, 2006.
- [11] R. Cerdan, R. Gilabert, and E. Vidal-Abarca. Selecting information to answer questions: Strategic individual differences when searching texts. *Learning and Individual Differences*, 21(2):201- 205, 2011.
- [12] N. Cliff. Dominance statistics: Ordinal analyses to answer ordinal ques-tions. *Psychological Bulletin*. 114(3):494–509, 1993
- [13] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-adapted Interaction*, 4(4):253-278, 1994.
- [14] J. Cromley and R. Azevedo. Testing and refining the direct and inferential mediation model of reading comprehension. *Journal of Educational Psychology*, 99(2):311-325, 20
- [15] A. Dinno. Package dunn.test. <https://cran.r-project.org/web/packages/dunn.test/dunn.test.pdf>, 2017.
- [16] E. Farhana, T. Rutherford, and C.F. Lynch. Associations Between Self-Regulated Learning Strategies and Science Assignment Score in a Digital Literacy Platform. In *Proceedings of the International Conference of the Learning Sciences*, 2020 (In Press).
- [17] E. Fouh, M. Farghally, S. Hamouda, K. H. Koh, and C.A. Sha er. Investigating di cult topics in a data structures course using item response theory and logged data analysis. *International Educational Data Mining Society*, 2016.
- [18] J. T. Guthrie and A. Wig eld. How motivation fits into a science of reading. *Scientific Studies of Reading*, 3(3):199-205, 1999.
- [19] R. K. Hambleton, H. Swaminathan, and H. J. Rogers. Fundamentals of item response theory. Sage, 1991
- [20] Y. Hosoda and D. Aline. Two preferences in question-answer sequences in language classroom context. *Classroom Discourse*, 4(1):63-88, 2013.
- [21] T. Hsieh and T. Wang. A mining-based approach on discovering courses pattern for constructing suitable learning path. *Expert Systems with Applications*, 37(6):4156-4167, 2010.
- [22] Y.-S. Hsu, M.-H. Yen, W.-H. Chang, C.-Y. Wang, and S. Chen. Content analysis of 1998-2012 empirical studies in science reading using a self-regulated learning lens. *International Journal of Science and Mathematics Education*, 14(1):1-27, 2016.
- [23] K. N. Llewellyn. *The Bramble Bush: On Our Law and its Study*. Oceana Publications, New York, 1960.
- [24] T. Makany, J. Kemp, and I. E. Dror. Optimising the use of note-taking as an external cognitive aid for increasing learning. *British Journal of Educational Technology*, 40(4):619-635, 2009.
- [25] D. McNamara. *Reading Comprehension Strategies: Theories, interventions, and technologies*. Psychology Press., 2007.
- [26] T. Michalsky. Integrating skills and wills instruction in self-regulated science text reading for secondary students. *International Journal of Science Education*, 35(11):1846-1873, 2013.
- [27] H. F. O'Neil Jr and R. S. Brown. Differential effects of question formats in math assessment on metacognition and affect. *Applied measurement in Education*, 11(4):331-351, 1998.
- [28] U. Pado. Question Difficulty- How to estimate without norming, how to use for automated grading. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1-10, 2017.
- [29] G. Pallier, R. Wilkinson, V. Danthiir, S. Kleitman, G. Knezevic, L. Stankov, and R. D. Roberts. The role of individual differences in the accuracy of confidence judgments. *The Journal of General Psychology*, 129(3):257-299, 2002.
- [30] Z. A. Pardos and N. T. Heffernan. Kt-idem: Introducing item di culty to the knowledge tracing model. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 243-254. Springer, 2011.
- [31] P. R. Pintrich. The role of goal orientation in self-regulated learning. In *Handbook of Self-regulation*, pages 451-502. Elsevier, 2000.
- [32] G. Rasch. Studies in mathematical psychology: I. probabilistic models for some intelligence and attainment tests. 1960.
- [33] G. A. Ravi and S. Sosnovsky. Exercise difficulty calibration based on student log mining. In *Proceedings of DAILE*, 2013.
- [34] Reading_AL. <https://www.activelylearn.com/post/infograph-ic-close-reading-strategies-with-actively-learn>
- [35] J. S. Richardson, R. F. Morgan, and C. Fleener. *Reading to Learn in the Content areas*. Cengage Learning., 2012.
- [36] D. Rizopoulos. ltm: An r package for latent variable modeling and item response theory analyses. *Journal of Statistical Software*, 17(5):1 - 25, 2006.
- [37] T. Rutherford. Within and between person associations of calibration and achievement. *Contemporary Educational Psychology*, 49:226-237, 2017.
- [38] L. Shen et al. Computer technology and college students' reading habits. *Chia-Nan Annual Bulletin*, 32:559-572, 2006.
- [39] S. Sosnovsky, P. Brusilovsky, D. H. Lee, V. Zadorozhny, and X. Zhou. Re-assessing the value of adaptive navigation

- support in e-learning context. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 193-203. Springer, 2008.
- [40] M. Taub, R. Azevedo, A. E. Bradbury, G. C. Millar, and J. Lester. Using sequence mining to reveal the efficiency in scientific reasoning during stem learning with a game-based learning environment. *Learning and Instruction*, 54:93-103, 2018.
- [41] K. Vitomir, D. Gasevic, S. Dawson, S. Joksimovic, R. S. Baker, and M. Hatala. Penetrating the black box of time-on-task estimation. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 184-193, 2015.
- [42] G. Weber and P. Brusilovsky. Elm-art: An adaptive versatile system for web-based instruction. *International Journal of Artificial Intelligence in Education*, 12:351-384, 2001.
- [43] C. E. Weinstein, J. Husman, and D. R. Dierking. Self-regulation interventions with a focus on learning strategies. In *Handbook of Self-regulation*, pages 727-747. Elsevier, 2000.
- [44] E. Wenger. *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kaufmann, 2014.
- [45] P. Winne, J. Nesbit, I. Ram, Z. Marzouk, S. D. Vytasek, J. and J. Stewart. Tracing metacognition by highlighting and tagging to predict recall and transfer. *AERA Online Paper Repository*. 2017.
- [46] P. H. Winne and A. F. Hadwin. Studying as self-regulated learning.. The educational psychology series. *Metacognition in Educational Theory and Practice*, 1998.
- [47] P. H. Winne and A. F. Hadwin. The weave of motivation and self-regulated learning. *Motivation and Self-regulated Learning: Theory, Research, and Application.*, 2008.
- [48] P. H. Winne and N. E. Perry. Measuring self-regulated learning. In *Handbook of Self-regulation*, pages 531-566. Elsevier, 2000.
- [49] M.-H. Yen, C.-Y. Wang, W.-H. Chang, S. Chen, Y.-S. Hsu, and T.-C. Liu. Assessing metacognitive components in self-regulated reading of science texts in e-based environments. *International Journal of Science and Mathematics Education.*, 16(5):797-816, 2018.
- [50] B. J. Zimmerman. Attaining self-regulation: A social cognitive perspective. In *Handbook of Self-regulation*, pages 13-39. Elsevier, 2000.
- [51] B. J. Zimmerman. Self-efficacy: An essential motive to learn. *Contemporary Educational Psychology*, 25(1):82-91, 2000.
- [52] B. J. Zimmerman and A. Bandura. Impact of self-regulatory influences on writing course attainment. *American Educational Research Journal*, 31(4):845-862, 1994.
- [53] B. J. Zimmerman, A. Bandura, and M. Martinez-Pons. Self-motivation for academic attainment: The role of self-efficacy beliefs and personal goal setting. *American Educational Research Journal*, 29(3):663-676, 1992.

Are You Really A Team Player? Profiling of Collaborative Problem Solvers in an Online Environment

Carol Forsyth
Educational Testing Service
90 New Montgomery, Ste. 1500
San Francisco, CA 94105
+1(415) 645-8465
cforsyth@ets.org

Jessica Andrews-Todd
Educational Testing Service
660 Rosedale Rd.
Princeton, NJ 08540
+1(609) 734-5809
jandrewstodd@ets.org

Jonathan Steinberg
Educational Testing Service
660 Rosedale Rd.
Princeton, NJ 08540
+1(609) 734-5324
jsteinberg@ets.org

ABSTRACT

Collaborative problem solving (CPS) is considered a necessary skill for students and workers in the 21st century as the advent of technology requires more and more people to frequently work in teams. In the current study, we employed theoretically-grounded data mining techniques to identify four profiles of collaborative problem solvers interacting with an online electronics task. The profiles were created based on 11 theoretically-grounded CPS skills defined a priori. The resulting four profiles correlated in expected directions with in-task performance and had interesting relationships with external measures associated with prior knowledge and CPS skills. These results inform and partially replicate findings from our previous research using a similar approach on a smaller dataset. Implications and comparisons between the two studies will be discussed.

Keywords

Collaborative Problem Solving, Ontology, Assessment, Simulation-based Assessment, Discourse

1. INTRODUCTION

With the increasing need for technology in workplace contexts, collaborative problem solving (CPS) is considered an important 21st century skill as workers are often required to complete complex tasks in teams to solve complicated, often technical problems. Accordingly, the need to teach and assess CPS has gained increased attention by researchers [4,5]. In research seeking to assess or teach CPS skills, researchers often employ digital technologies to capture evidence and improve assessment of CPS, as this skill is complex and includes many facets.

In defining the facets of CPS there is little dispute that the construct includes social and cognitive dimensions [1,22]. The social dimension is meant to be interpersonal, including features such as sharing information and perspective taking. These types of features are associated with building a shared understanding among team

members which is essential for building common ground, an important component of completing a task [6]. The cognitive dimension includes components such as planning, representing the problem, and formulating hypotheses. These components are complex in nature and therefore difficult to assess with traditional assessments such as multiple-choice questions without compromising fidelity and generalizability [7]. Therefore, assessment researchers have turned to online environments, including games and simulations, which allow for collaboration among team members to capture the discourse and complex actions necessary to evaluate CPS competency.

To evaluate CPS competency in online environments, both a competency model and advanced analytic techniques are often needed. Specifically, a competency model is necessary to identify skills and features aligning to specific constructs. Analytic approaches are needed to deal with the large streams of data stored in log files while also accounting for the underlying competency model and theoretical explanations [12].

Accordingly, in an effort to assess students' CPS skills in an online environment, we employ a theoretically-grounded data mining approach [9] incorporating a conceptual model and machine learning approaches in an iterative process. Specifically, we define a competency model based on existing literature that identifies features a priori. Our competency model is based on our prior work [1,2,3] and used to extract features in a meaningful way, and machine learning algorithms are used to profile students. We then interpret and refine algorithms based on theoretical interpretations. Thus, the process is a collaborative effort between computer scientists, learning scientists, psychometricians, and cognitive psychologists. In the current study, this principled process is used to replicate findings from a previous study [2] by discovering profiles of collaborative problem solvers that are strongly grounded in theory associated with cognitive and social psychological research. We then validate these profiles with external measures and compare these profiles with our previous findings from students interacting with the same online collaborative electronics task.

2. METHODS

2.1 Participants

Students in electronics, engineering, and physics programs were recruited from universities and community colleges across the United States to complete the study. In total, there were 378 students who participated. Of those students who reported their

Carol Forsyth, Jessica Andrews-Todd and Jonathan Steinberg
"Are You Really a Team Player?: Profiles of collaborative problem solvers in an online environment" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 403 - 408

gender, 76% were males and 21% were females with 3% other, preferring not to respond, or unreported. Of those who reported their race, 62% were White, 7% were Black or African American, 8% were Asian, 10% reported being more than one race, 1% reported Other, with 4% preferring not to answer or unreported. For ethnicity, 7% reported being Hispanic. The modal age range among students was 18 to 20 years old.

2.2 Tasks and Measures

To complete the study students first completed a pretest about electronics concepts to gather information about their content knowledge, next progressed to the online electronics task, and then completed self-report measures where they rated themselves and their teammates on CPS capabilities along social and cognitive dimensions. We will first discuss the external measures and then the online electronics task.

2.2.1 External Prior Knowledge Test

The external prior knowledge test was created by a group of experts concerning the series circuit problem. First a conceptual map of the problem was created. Then, a q matrix defining skills and complexity was devised to create an equal number of questions for each electronics skill necessary to solve the series circuit problem. Next, the final items were validated by experts as well as through psychometric analysis. As a result, the original test included 28 items but only 23 saliently reflected the original intent of the test developers based on a CFA [24]. Thus, the total score per student for the 23 items is the measure of prior knowledge.

2.2.2 CPS Inventory

The CPS Inventory serves as a self-report measure of CPS skills that aligns to a competency model of CPS (which will be discussed in more detail in the next section). The Inventory consists of 14 items, seven of which correspond to social CPS behaviors (e.g., I tried to establish a good relationship with my teammates) and seven of which correspond to cognitive CPS behaviors (e.g., I helped develop a plan to solve the problem). There is a “self” version of the Inventory where participants rate their own CPS behaviors on a 4-point Likert scale (1=strongly disagree, 4=strongly agree) and a “team” version where participants rate their team’s CPS behaviors as a whole on a 4-point Likert scale (1=strongly disagree, 4=strongly agree). The CPS Inventory was administered after students completed the electronics task described next.

2.2.3 Three-Resistor Activity

Students solved a collaborative problem on electronics concepts associated with Ohm’s Law and Kirchhoff’s Voltage Law. Each student in a team of three worked on a separate computer, each running a simulation of an electronics circuit. Each student’s circuit was connected to form a series circuit.

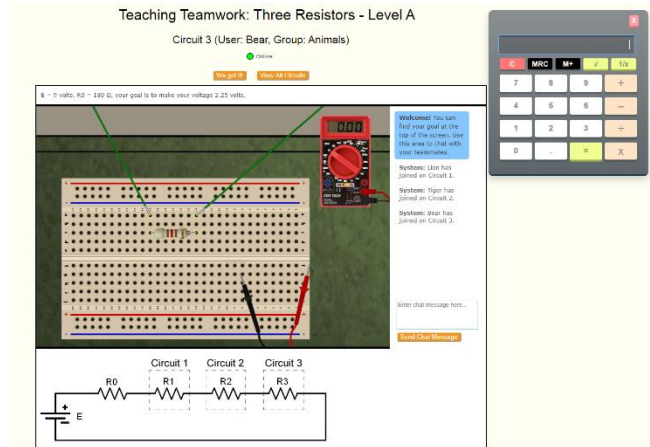


Figure 1. Screenshot of Three-Resistor Activity

Students were randomly assigned into teams by their instructors and team members were anonymized with provided usernames (e.g., Lion, Tiger, Bear). Within the interface, each student had a chat window, a digital multi-meter (DMM), probes extending from the DMM (red and black), a resistor, a zoom button, a calculator, and a submit button (See Figure 1 for a screenshot of the task interface). These features of the interface made it possible for students to communicate with teammates, take measurements, view and change the resistance on their boards, zoom out to view other teammates’ boards, perform calculations, and submit answer choices.

In the task, students had the goal of reaching a specified goal voltage value on each of their circuit boards. Because each of the circuits were connected in series, any changes made on one circuit board would affect readings on all teammates’ circuits which required the need for collaboration around coordinating actions so that everyone could reach their goal voltage values. The task has four levels which increase in difficulty as one variable changes in the task interface. In Level 1 each student had the same goal voltage value to achieve and the values of the resistance (R_0) and supply voltage (E) of an external, fourth circuit in the series that students could not control were provided. In Level 2 the resistance and supply voltage of the external, fourth circuit were still provided, but each teammate now had a different goal voltage to reach. In Level 3, teammates had different goal voltages, the external resistance was provided, but the external supply voltage was unknown and needed to be found to solve the problem. In Level 4, teammates again had different goal voltages, but now both the external resistance and supply voltage were unknown and needed to be found. The task was designed so that students may only proceed to attempt the next level after completing the previous level. Therefore, levels attempted can be used as a proxy measure for performance on the electronics task. To identify the CPS skills exhibited while solving the Three-Resistor Activity, a CPS conceptual framework outlined in the form of an ontology was created, as described next.

2.2. CPS Conceptual Framework

A CPS ontology (similar to a concept map) was created using the In-Task Assessment Framework (I-TAF) approach [3,12]. This approach is an augmented version of evidence-centered design (ECD) that supports identification of features of complex constructs in online environments.

Creating the ontology required iterative refinement with the support of subject matter experts and data. The ontology was created based on literature from areas such as computer-supported collaborative learning, organizational psychology, individual problem solving, and linguistics [11,14,15, 18,19,20,21,22,23]. Data collected from the Three-Resistor Activity then informed changes to the ontology so that it most accurately reflects the construct as well as associated skills, strategies, tactics, and features based on real data collected from students interacting with the task.

To visually display the various components of CPS, the ontology is designed hierarchically. The construct (i.e., CPS) sits at the top with the two dimensions of CPS (social and cognitive) as second layer nodes. The social and cognitive nodes are linked to CPS skills associated with each dimension. Specifically, there are four skills in the social dimension and five skills in the cognitive dimension. The social dimension includes maintaining communication, sharing information, establishing shared understanding, and negotiating. The cognitive dimension includes exploring and understanding, representing and formulating, planning, executing, and monitoring. For a more in-depth discussion of this work, please refer to [1,2,3].

The nine high-level CPS skills are linked to 23 sub-skills on the fourth and fifth layers of the ontology. These sub-skills more explicitly define each of the nine CPS skills. For example, the sharing information CPS skill includes three sub-skills, sharing one's own information, sharing task or resource information, and sharing understanding. The sub-skills are connected to an evidence model which provides nodes corresponding to strategies or behaviors needed to indicate evidence of each sub-skill. The strategy nodes are then linked to tactic nodes which correspond to in-task affordances available to carry out a given strategy and subsequently feature nodes that can be inferred from individuals' behaviors. These features are identified in the log files for extraction and additional analysis. See Figure 2 for an example of the structure of a portion of the CPS ontology.

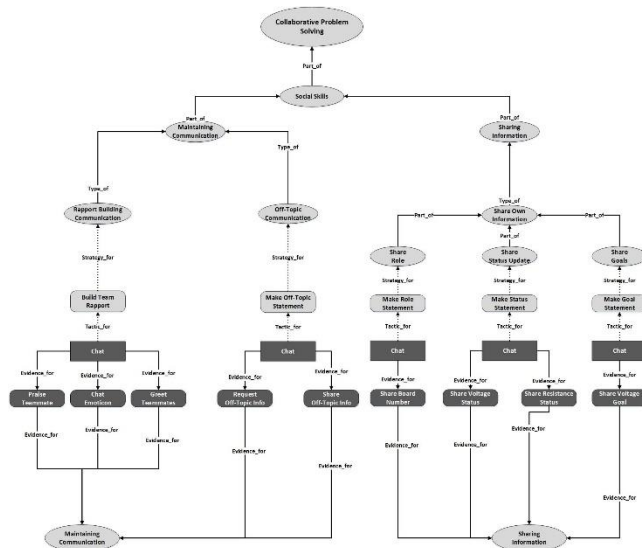


Figure 2. CPS Ontology Fragment

For this particular ontology, the majority of the skills are represented by discourse features associated with team members chatting amongst each other. As there are limited natural language processing tools to identify such low level and abstract features of CPS, qualitative coding was conducted.

2.3 Qualitative Coding

The qualitative coding was conducted on 51,805 rows of log data corresponding to student-generated chats and actions (e.g., resistor changes, calculations) to identify the 23 CPS sub-skills. Specifically, three raters coded each log file event, with each event only receiving one code. To examine inter-rater reliability, a random sample of 20% of the data were triple coded. The inter-rater reliability among the raters was found to be Kappa = .93, indicating substantial agreement [13]. All discrepancies among the coders were discussed to reach consensus for a final code. The remaining data were coded individually by the raters.

During qualitative coding, raters were looking for evidence of one of 23 sub-skills within nine high-level CPS skills under the social and cognitive dimensions of CPS. We next describe each of the sub-skills in turn. In the social dimension, maintaining communication corresponds to content-irrelevant social communication [15,16] and includes three sub-skills, rapport building communication (e.g., greeting teammates), off-topic communication (e.g., talking about homework from another class), and inappropriate communication (e.g., denigrating teammates). Sharing information corresponds to content-relevant information used in the service of solving the problem [22,25] and includes three sub-skills, sharing one's own information (e.g., sharing one's goal voltage), sharing task or resource information (e.g., sharing where the calculator is located in the task interface), and sharing the state of one's understanding (e.g., metacognitive statements such as, "I don't know"). Establishing shared understanding corresponds to communication used to learn the perspective of others and ensure that communication is understood by others [6]. This CPS skill includes two sub-skills, the presentation phase (e.g., requests for information) and the acceptance phase which includes responses indicating comprehension or lack of comprehension of a statement. Negotiating corresponds to communication used to identify conflicts and resolve those that arise [11], and includes three sub-skills, expressing agreement (e.g., "you are right"), expressing disagreement (e.g., "that's not right"), and resolving conflicts.

In the cognitive dimension, exploring and understanding corresponds to actions used to explore the task interface and understand the problem [21] and includes two sub-skills, exploring the environment (e.g., spinning the dial on the DMM) and understanding the problem. Representing and formulating corresponds to communication used to build a mental representation of the problem and formulate hypotheses for how to solve the problem [19,21]. There are two sub-skills for this CPS skill, representing the problem (e.g., "this is a series circuit") and formulating hypotheses (e.g., "I think if everyone has 470 ohms it will be 3.25"). Planning corresponds to communication used to develop a strategy for solving the problem [11, 21], and includes three sub-skills, setting goals (e.g., "We need 6.69 V across our resistors"), managing resources (e.g., "We need to find numbers and decide who does what"), and developing strategies (e.g., "Let's find E first using Kirchhoffs voltage law"). Executing corresponds to communication and actions used in the service of carrying out a plan [21]. This CPS skill includes three sub-skills, enacting strategies (e.g., performing calculations), directing actions (e.g., "Adjust yours to 300 ohms"), and reporting actions (e.g., "I set mine to 120"). Monitoring corresponds to communication and actions in the service of monitoring teammates or progress toward the goal [21,22], and includes two sub-skills, monitoring team organization (e.g., checking on the status of teammates or clicking the Zoom button) and monitoring success (e.g., "We got it" or clicking submit).

3. ANALYSES AND RESULTS

After completing the qualitative coding, the quantitative analyses were conducted in two stages: profile discovery and then validation. For the profile discovery, we performed a hierarchical cluster analysis on the frequencies of each individual's display of the high-level CPS skills using the Ward method [26] as this was an appropriate clustering method given the sample size [17]. We collapsed the 23 sub-skills into the high-level CPS skills for the cluster analysis in order to replicate the process used in previous research [2]. Next, the revealed profiles were compared according to their task performance as identified by number of task levels attempted, performance on the electronics pre-test, and ratings on the self and team CPS Inventory with Kruskal-Wallis tests and Mont Carlo simulations to ensure accurate statistical significance.

3.1 Profile Discovery

We discovered profiles of various types of collaborative problem solvers based on the CPS skills as determined by the competency model (i.e., CPS ontology). Since two of the CPS skills (monitoring and executing) each had both chat and actions as features to determine these skills, we separated them into separate chat and action skills (i.e., monitoring chats, monitoring actions, executing chats, executing actions). Thus, the total number of CPS skills clustered were 11. A hierarchical cluster analysis using the Ward method [26] was conducted on the standardized frequencies of CPS skills displayed for each student. The final number of resulting profiles was determined based on a theoretical interpretation of each of the profiles. Therefore, the profiles were not chosen by fit metrics alone but rather how meaningful these profiles were with respect to social and cognitive psychological research. This was a similar approach to that which was used in our prior work [2]. Although the method was similar, the resulting profiles had some differences. Four profiles emerged with varying sample sizes, which were named Social Loafers, Active Collaborators, Super Socials, and Low Collaborators. In our interpretation of the profiles, we used standardized average frequencies of the CPS skills to discuss patterns across the four profiles.

3.1.1 Social Loafers

The Social Loafers ($n = 190$) were a group of individuals that displayed below average frequencies of every CPS skill. These individuals did not contribute much to the team's problem solving. Social loafing has a long history in social psychology as a phenomenon where individuals assume that other team members will complete the task and therefore reduce their own effort [14].

3.1.2 Active Collaborators

The Active Collaborators ($n = 24$) displayed high frequencies on all of the identified CPS skills in the competency model [3] except for monitoring actions ($z = -.28$). Indeed, these individuals had z values greater than 1 on two of the CPS skills, and greater than 2 on another two CPS skills consistently indicating students being on average, at least an entire standard deviation above the overall mean. Specifically, sharing information ($z = 1.44$) and establishing shared understanding ($z = 1.08$), were above the mean. Furthermore, executing chats ($z = 2.23$) and monitoring chats ($z = 2.83$) were over a standard deviation above the mean. All other CPS skills had positive standardized values, indicating that these students were generally active in communicating with teammates and helping solve the problem.

3.1.3 Super Socials

The Super Socials ($n = 91$) showed high frequencies on the social dimension of CPS skills [1], but lower frequencies for the

cognitive CPS skills in comparison (except for representing and formulating). Specifically, these individuals showed the highest demonstration of negotiating behaviors in comparison to the other profiles ($z = 1.02$) and had positive standardized values on all other social skills, though not quite at the level of the Active Collaborators. The only cognitive CPS skills with positive standardized values were communication-based behaviors representing and formulating ($z = 1.08$), planning ($z = .51$), executing chats ($z = .21$), and monitoring chats ($z = .09$).

3.1.4 Low Collaborators

The Low Collaborators profile ($n = 73$) consisted of individuals that did not appear to collaborate with their teammates based on the features in the competency model [3]. However, they did show high levels of action-based cognitive behaviors including exploring and understanding ($z = .77$), monitoring actions ($z = 1.21$) and executing actions ($z = .76$). The Low Collaborators had negative standardized values for all other CPS skills. These individuals appeared to work alone without communicating with their teammates which is different from the Social Loafers who simply did not do much work at all.

3.2 Profile Validation

The profiles were validated with both log data performance metrics as well as external measures.

3.2.1 In-Task Performance and Profile Membership

There was a significant relationship between profile membership and the number of task levels attempted, a proxy for performance in the task, ($X^2(3,370) = 7.66, p = .05, \text{partial } \eta^2 = .02$). The Monte Carlo simulation for significance with 10,000 samples revealed a significance level of $p = .05$ (lower bound $p = .047$, upper bound $p = .059$). Specifically, the mean ranks, where higher values corresponded to more levels attempted, were the lowest for the Social Loafers (175.00) and highest for the Active Collaborators (219.88) which is similar to our previous findings [2]. The mean ranks for the Super Socials and Low Collaborators fell in between the aforementioned profiles (194.87 and 189.57, respectively). These patterns indicate that the Active Collaborators and Super Socials had higher mean ranks on performance than Social Loafers and Low Collaborators.

3.2.2 Pre-Test Performance and Profile Membership

The profiles were compared to the external electronics pre-test, a measure of prior knowledge. The test included 23 items that were summed to create a score for each student participant. Results revealed that there was a significant relationship between profile membership and performance on the electronics test ($X^2(3, 370) = 8.83, p < .05, \text{partial } \eta^2 = .02$). The Monte Carlo simulation for significance with 10,000 samples revealed a significance level of .027 (lower bound $p = .021$, upper bound $p = .031$). The highest mean rank for prior knowledge was for the Active Collaborators (212.73) and the lowest was for the Low Collaborators (172.10). Ranging in the middle, the Super Socials had higher mean ranks than the Social Loafers (209.42 and 175.45, respectively). Post hoc comparisons with a Bonferroni correction revealed a marginally significant difference between Social Loafers and Super Socials ($p = .08$). No other pairwise comparisons approached statistical significance (all p 's $> .10$).

3.2.3 Post-Task Self-Report and Profile Membership

The profiles were compared to student's ratings of their own CPS behaviors (Self CPS Inventory) and their team's CPS behaviors (Team CPS Inventory).

There was a significant relationship between self-ratings of CPS skills (sum of ratings for Self CPS Inventory) and cluster membership ($X^2(3,349) = 15.57, p < .05, \text{partial } \eta^2 = .05$). The Monte Carlo simulation with 10,000 samples revealed a significance level of $p = .001$ (lower bound $p = .001$, upper bound $p = .002$). Mean ranks were highest for the Super Socials (210.18) and lowest for the Social Loafers (160.58), with the Active Collaborators having higher mean ranks than the Low Collaborators as expected (184.96 and 162.22, respectively). Post hoc comparisons with a Bonferroni correction revealed a significant difference between Low Collaborators and Super Socials ($p < .02$) and Social Loafers and Super Socials ($p = .001$).

There was a significant relationship between ratings on the Team CPS Inventory and profile membership as well ($X^2(3,349) = 9.04, p < .05, \text{partial } \eta^2 = .03$). Monte Carlo simulation with 10,000 samples revealed a significance of $p = .028$ (lower bound $p = .024$, upper bound $p = .032$). The highest mean rank was for the Super Socials (199.47) and the lowest was for the Social Loafers (161.53), with Active Collaborators having higher mean ranks than Low Collaborators as expected (191.74 and 171.78, respectively). Post hoc comparisons with a Bonferroni correction revealed a significant difference between the Super Socials and Low Collaborators ($p = .02$).

4. CONCLUSIONS

Overall, we discovered four meaningful profiles of types of collaborative problem solvers: Social Loafers, Active Collaborators, Super Socials, and Low Collaborators. These profiles had significant relationships with in-task performance, electronics prior knowledge, and self-reported CPS capabilities.

The four profiles discovered partially replicate previous findings [2]. Specifically, in our previous study, Social Loafers and Active Collaborators also emerged as profile groups. The Social Loafers could also be called “Free Riders” as these individuals do not contribute much to solving the problem with their teammates. Conversely, the Active Collaborators, which were a small subset of the sample, performed well on all measured aspects of CPS. As expected, Active Collaborators showed better in-task performance than Social Loafers which replicates findings from our prior work [2]. This makes sense as the Active Collaborators displayed high frequencies of CPS behaviors and should therefore have performed well on the task. Social Loafers may have been expecting others to do the work and therefore should not have performed as well on the task.

There were two new profiles that differed but still augmented our previous findings. We attribute these differences to a change in sample size and its diversity. The sample size was nearly three times the size of the previous sample and included students in a wider variety of domains, including electronics, engineering, and physics. The new profiles that emerged in this experiment included the third profile called the Super Socials which does align with other profiles that have been examined. Specifically, in prior work we have found what we termed a high social/low cognitive profile that behaved similarly in displaying high levels of social CPS behaviors and comparatively lower levels of cognitive CPS behaviors. This profile was discovered by an examination of the two dimensions based on means of the CPS features rather than cluster analysis [1]. Beyond this, other work has found a profile designated as “Compensating Collaborators” who had high collaboration actions but performed poorly on problem solving variables [10]. The last profile, the Low Collaborators, also did not emerge in our prior cluster analysis work [2] but could be usefully compared to the Chatty Doers from that work. Similar to the Low

Collaborators, the Chatty Doers demonstrated a high level of executing actions, but in contrast, the Chatty Doers did engage in communication with their teammates, though most of the communication was in the maintaining communication category. Interestingly, the Low Collaborators did not seek to engage with their teammates and instead appeared to work alone by engaging in executing and exploratory actions.

In regards to prior knowledge, Active Collaborators and Super Socials demonstrated the first and second highest average scores on the electronics pre-test. It is possible that their higher prior knowledge enabled them to engage in more communication behaviors and problem-solving behaviors (in the case of the Active Collaborators) to contribute to solving the problem. The opposite could be said for the Social Loafers and Low Collaborators, the latter of which had the lowest average pre-test performance. For example, perhaps the Low Collaborators did not want to collaborate with others and preferred to work alone because they were embarrassed of their low levels of content knowledge. On the other hand, perhaps the Low Collaborators already had low content knowledge because of their refusal to work with, and therefore learn from, others on previous tasks. Causality and directionality certainly cannot be determined by these analyses. However, these findings do suggest that testing these hypotheses may provide important insights for CPS researchers.

The Self and Team CPS Inventories required students to rate themselves based on their own metacognitive judgments of their own CPS behaviors as well as their team’s CPS behaviors. The Super Socials had the highest ratings for CPS behaviors both for themselves and for their team while the Social Loafers had the lowest ratings on each inventory. These results were expected, though we would also expect high ratings for the Active Collaborators and lower ratings for the Low Collaborators (mean ranks showed such patterns).

All of these findings together suggest that further research should be conducted to explore whether the same kinds of patterns of results emerge with relationships among profiles such as the ones observed in this study, in-task performance, and other ratings. One limitation of this study is that the in-task performance measure includes aspects of the contributions from others while the CPS profile is based on an individual’s contributions. Despite the interdependent nature of the electronics task, we are continuing work in developing an alternative in-task performance measure that potentially incorporates only individual contributions. Furthermore, the CPS Inventory relies on self-judgments which can sometimes have biases [8]; however, we did want to incorporate some external measure of CPS behaviors that could be compared to participants’ in-task CPS behaviors. Finally, the CPS skills used to develop the profiles included only the higher aggregate level CPS skills, as the sample size was not sufficient to include the lower level coded data.

Overall, we found that this study, which included a larger sample size and new external measures relative to our previous work, partially replicated and informed our previous findings. Our theoretically-grounded data mining approach appears to reveal meaningful profiles on two separate data sets with students completing the same electronics task. We hope that this work will inform future work on ways to incorporate theory and data-driven approaches to make inferences about individuals’ CPS capabilities, and contribute to a better understanding of types of collaborative problem solvers, including how certain CPS behaviors relate to various relevant measures.

5. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant DUE 1535224. The opinions expressed are those of the authors and do not necessarily represent views of the National Science Foundation.

6. REFERENCES

- [1] Andrews-Todd, J. and Forsyth, C. M. 2020. Exploring social and cognitive dimensions of collaborative problem solving in an open online simulation-based task. *Comp. in Human Beh.*, 104, (2020) 105-759.
- [2] Andrews-Todd, J., Forsyth, C. M., Steinberg, J., and Rupp, A. A. 2018. Identifying profiles of collaborative problem solvers in an online electronics environment. In K. E. Boyer & M. Yudelson (Eds.), *Proceedings of the 11th International Conference on Educational Data Mining* (Buffalo, NY July 15-18, 2018). EDM '18 International Educational Data Mining Society, 239–245.
- [3] Andrews-Todd, J. and Kerr, D. 2019. Application of ontologies for assessing collaborative problem solving skills. *Intern. Journ. of Testing*, 19, 2 (2019), 172–187.
- [4] Andrews, J. J. and Rapp, D. N. 2015. Benefits, costs, and challenges of collaboration for learning and memory. *Trans. Issues in Psych. Sci.*, 1, 2 (2015), 182-191.
- [5] Burrus, J., Jackson, T., Xi, N., and Steinberg, J. 2013. Identifying the most important 21st century workforce competencies: An analysis of the occupational Information network (O*NET). ETS Research Report RR-13-21 (2013). Educational Testing Service, Princeton, NJ.
- [6] Clark, H. H. 1996. *Using Language*. Cambridge University Press, New York, NY.
- [7] Davey, T., Ferrara, S., Shavelson, R., Holland, P., Webb, N., and Wise, L. 2015. *Psychometric considerations for the next generation of performance assessment*. 2015. Educational Testing Service, Princeton, NJ.
- [8] Dunlosky, J. and Metcalfe, J. (2007). *MetaCognition: A Textbook for Cognitive, Educational, Life Span & Applied Psychology*. Sage Publications, Los Angeles, CA.
- [9] Forsyth, C.M., Graesser, A. C., Pavlik, P., Millis, K., and Samei, B. 2014. Discovering theoretically grounded predictors of shallow vs. deep- level learning. In Stamper, J., Pardos, Z., Mavrikis, M., McLaren, B.M., Eds. *In Proceedings of the 7th International Conference on Educational Data Mining*. (London, U.K. July 4- 7, 2014) EDM '14. International Educational Data Mining Society, 229-232.
- [10] Herborn, K., Mustafić, M., and Greiff, S. 2017. Mapping an experiment-based assessment of collaborative behavior onto collaborative problem solving in PISA 2015: A cluster analysis approach for collaborator profiles. *Journ. of Educ. Measur.*, 54, 1, 103–122.
- [11] Hesse, F., Care, E., Buder, J., Sassenberg, K., and Griffin, P. 2015. A framework for teachable collaborative problem solving skills. In P. Griffin and E. Care, Eds. *Assess. and teaching of 21st century skills*. Springer, New York, NY, 37–56.
- [12] Kerr, D., Andrews, J. J., and Mislevy, R. J. 2016. The in-task assessment framework for behavioral data. In A. A. Rupp and J. P. Leighton, Eds. *The handbook of cognition and assessment: Frameworks, methodologies, and applications*. Wiley-Blackwell, Hoboken, NJ, 472-507.
- [13] Landis, J. R. and Koch, G. G. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33, 1 (1977), 159–174.
- [14] Latané, B., Williams, K., and Harkins, S. 1979. Many hands make light the work: The causes and consequences of social loafing. *Journ. of Pers. and Soc. Psycho.* 37, 6 (1979), 822-832.
- [15] Lipponen, L. (2000). Towards knowledge building: From facts to explanations in primary students' computer mediated discourse. *Learn. Environ. Res.*, 3, 2 (2000), 179–199.
- [16] Liu, L., von Davier, A. A., Hao, J., Kyllonen, P., and Zapata-Rivera, J.-D. 2015. A tough nut to crack: Measuring collaborative problem solving. In Y. Rosen, S. Ferrara, and M. Mosharraf, Eds. *Handbook of research on computational tools for real-world skill development*, IGI-Global, Hershey, PA, (2015) 344–359.
- [17] MacQueen, J. B. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. (1967). University of California Press, Berkeley, CA, 281-297.
- [18] Meier, A., Spada, H., and Rummel, N. 2007. A rating scheme for assessing the quality of computer-supported collaboration processes. *Inter. Journ. of Computer-Supported Collab. Learning*, 2, 1 (2007), 63–86.
- [19] Mayer, R. E. and Wittrock, M. C. 1996. Problem-solving transfer. In D. C. Berliner & R. C. Calfee (Eds.), *Handbook of educational psychology* (1996) Lawrence Erlbaum, 47–62.
- [20] Morgan, B. B., Salas, E., and Glickman, A. S. 1993. An analysis of team evolution and maturation. *Journal of Gen. Psych.*, 120, 3 (1993), 277–291.
- [21] OECD. 2013a. *PISA 2012 assessment and analytical framework: Mathematics, reading, science, problem solving and financial literacy*. OECD Publishing, Paris.
- [22] OECD. 2013b. *PISA 2015 collaborative problem solving framework*. OECD Publishing, Paris.
- [23] O'Neil, H. F., Chuang, S., and Chung, G. K. W. K. 2003. Issues in the computer-based assessment of collaborative problem solving. *Assessment in Education: Principles, Policy & Practice*, 10, 3 (2003), 361–373
- [24] Steinberg, J., Andrews-Todd, J., Forsyth, C., Chamberlain, J., Horwitz, P., Koon, A., Rupp, A. A., and McCulla, L. 2019. *The Development of a Content Assessment of Basic Electronics Knowledge*. Paper presented at 50th Annual Conference of the Northeastern Educational Research Association, Trumbull, CT. (October, 2019).
- [25] van Boxtel, C., van der Linden, J., and Kanselaar, G. 2000. Collaborative learning tasks and the elaboration of conceptual knowledge. *Learn. and Instruc.*, 10, 4 (2000), 311–330.
- [26] Ward, Jr., J. H. 1963. Hierarchical grouping to optimize an objective function. *Journ. of the American Stat. Assoc.*, 58, 301 (1963), 236-244. DOI: 10.2307/2282967

Student Teamwork on Programming Projects

What can GitHub logs show us?

Niki Gitinabard, Ruth Okoilu, Yiqiao Xu, Sarah Heckman, Tiffany Barnes, & Collin Lynch
North Carolina State University
{ngitina, rookoilu, yxu35, sarah_heckman, tmbarnes, cflynch}@ncsu.edu

ABSTRACT

Teamwork, often mediated by version control systems such as Git and Apache Subversion (SVN), is central to professional programming. As a consequence, many colleges are incorporating both collaboration and online development environments into their curricula even in introductory courses. In this research, we collected GitHub logs from two programming projects in two offerings of a CS2 Java programming course for computer science majors. Students worked in pairs for both projects (one optional, the other mandatory) in each year. We used the students' GitHub history to classify the student teams into three groups, *collaborative*, *cooperative*, or *solo-submit*, based on the division of labor. We then calculated different metrics for students' teamwork including the total number and the average number of commits in different parts of the projects and used these metrics to predict the students' teamwork style. Our findings show that we can identify the students' teamwork style automatically from their submission logs. This work helps us to better understand novices' habits while using version control systems. These habits can identify the harmful working styles among them and might lead to the development of automatic scaffolds for teamwork and peer support in the future.

Keywords

collaborative learning, version control, study habits, secondary education, GitHub, team projects

1. INTRODUCTION

Teamwork is an essential component of professional software development and CS educators incorporate it into their curricula to better prepare students for future careers [12, 34]. Working in teams provides students the opportunity to work on larger-scale projects than they otherwise would, and is more consistent with industry practice. Team projects also allow students to learn from their peers as described by *Social Learning Theory* (SLT) [3]. *Social Learning Theory* highlights four principal requirements for learning in social environments - **attention** or the opportunity to observe each other's work, **reproduction** or the chance to implement what they learned from observations, **retention** or being continuously engaged in the team, and **motivation** for learning [3].

Niki Gitinabard, Ruth Okoilu, Yiqiao Xu, Sarah Heckman, Tiffany Barnes and Collin Lynch "Student Teamwork on Programming Projects. What can GitHub logs show us?" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 409 - 416

Prior research suggests that having all the team members engaged in the project is essential for success and for student learning. Seers et al. showed that the balance of contributions in a team is correlated with team performance and member satisfaction [36]. Chen et al. argued that uneven teamwork, where one member does the majority of the work, may limit the learning opportunities for their peers as well as themselves [9]. Further, students that do not contribute may become less motivated to make genuine efforts since they can rely on their teammates to pick up the extra work [40].

Many researchers have measured and studied effective coding and study habits for individuals [2, 38, 37, 22, 41, 6, 5, 42, 16, 8]. However, evaluating the quality of students' teamwork is more complicated. Hoegl et al. defined teamwork quality metrics as communication, coordination, balance of member contribution, mutual support, effort, and cohesion [15]. Most of these metrics are not easy to quantify. Some have relied on surveys [10] or supervisor assessments [28] for the evaluation of students' teamwork process, but little work has been done evaluating student teamwork quality and contribution in CS secondary education programming projects based on online activities [31]. As a result, in this work we aim to automatically identify the teams with weaker teamwork styles.

Teamwork in software development projects can be described in three ways. Coman et al. address two forms of teamwork: "Collaborative", where the teammates share the same goal toward solving an issue (in their case sharing a programming task), and "Cooperative", where they support each other while working on different goals [11]. We observed that students took similar approaches in our group coding assignments. Some worked on similar parts of the project (e.g. both working on implementation or both testing, etc.) at the same time. Since each of these parts were focused on a specific goal (e.g. implementation adds program features and writing test cases improves code coverage and finds issues), working on the same parts means having a similar goal and these teams are similar to Collaborating teams. In such teams all the members have significant contributions to the same parts and they might even do pair-programming at times. Other groups of students divided the work by the project part (e.g. one works mostly on implementation and the other works on testing) while all contributing significantly. They might assist each other when necessary, but most of the work in each part was done by one member, focusing on the specific goal of each part. This behavior is similar to Cooperative work as mentioned by Coman et al.. The third form of teamwork is having a *free-rider*, which as mentioned by van der Duim, is common in group projects [40]. We refer to this form of teamwork as "Solo-submitting" where one member did most of the work. The other members might have a few commits where they made a quick fix, but the majority of work was done by a single member.

In this work, we analyzed data collected from two offerings of a CS2 course on Java programming for CS majors. In this course, students must complete three programming projects, one independent assignment, one where pairing is optional, and one where it is mandatory. The students are required to use GitHub as a version control system and for assignment submissions. Also, their GitHub repositories are connected to a Jenkins [1] server, which provides them with responses from instructor-defined unit tests every time they submit their code. We analyzed student commit behaviors to define metrics for evaluating their contributions to the team and to classify their work style. We tagged 400 commit messages as referring to different parts of the student projects (i.e. Implementation, Testing, Bugfix, Merge, Documentation, Style, and Other) that were graded in this course. We then used natural language processing to learn from that sample and tag the remaining commit messages. We finally used the commits in different categories to define several metrics for students' contributions to the team such as the *Number of implementation commits* and the *Percentage of testing*. In order to obtain a ground truth metric for the teamwork style we engaged two subject matter experts to classify 100 of the 238 team repositories in these classes into one of the three categories: "Collaborative", "Cooperative", and "Solo-submit". Then, we used the metrics to train and test prediction models on the teamwork styles of these projects.

To be more specific, we test the following set of hypotheses:

- H1. We can automatically classify commit messages into different parts of the project.
- H2. We can automatically classify student teams into Collaborative, Cooperative, or Solo-submit.

The findings of this study will help us to develop metrics to evaluate the effectiveness of student project teams and eventually provide students adaptive guidance or flag teams for instructor intervention.

2. BACKGROUND

Prior researchers have analyzed students' work on programming projects with the goal of identifying good habits that are common to higher performers and bad habits that are not [2, 38, 37, 22, 41, 6, 7, 5, 42, 16, 8]. Some researchers have also used visualization tools to analyze students' activity patterns and to present guidance to the students themselves (e.g. Retina [26, 18]). One more recent approach to analyzing students' behavior is based upon studying logs from version control systems [14, 34, 26, 25]. However, prior studies in this area have primarily focused on the students' individual work habits and not on the role that they play in a team. While other researchers have studied teamwork in CS courses (e.g [27, 39, 4, 30, 43, 12, 19]), these studies have generally relied upon student surveys and evaluations to bound their performance and only a few have considered their online behavior [18, 23, 13]. Thus, there is little prior work on detailed analyses of how individual student features affect team performance.

While teamwork is the norm in industry, students may be unfamiliar with norms of collaborative work and many things can go wrong in team projects [12]. For example, some team members may decide to "gang up" and leave others out of the decisions or they might decide to be "free-riders" and do no work at all [35, 40]. A number of researchers have studied the impact of teamwork on student performance and ways to enhance the experience of collaborative class work. Higher performing students often believe that they worked with greater initiative than their teammates, mostly alone, and they tend to give up on collaborative work [20]. Additionally, there are users who prefer to work alone, mostly

called "lone wolves" and their inclusion in teams often has a negative impact on the team's overall performance on the project [4]. Instructors could use online contributions to easily identify some of these harmful patterns.

Another use for evaluating student contributions is to measure their teamwork quality. Most of the prior studies in classes evaluate the quality of the teamwork and their satisfaction with the teamwork experience based on the students' final peer evaluations [39, 43, 12, 19, 20]. While peer evaluation is a popular method among the instructors and is often used for grading group work [30, 19], it can be difficult to calculate student grades using their peers' estimations of their share of work [12]. There are also other methods such as video-taping students while collaborating [27]. As suggested by Hoegl, the balance of students' contributions (i.e. having almost equal shares in the project) to the team is also an effective measure for team quality [15]. Seers et al. also mentioned that the balance in the team members' contributions is related to team performance [36]. However, measuring member contributions to software projects is not easy.

Other approaches have also been proposed for measuring team member contributions. One method relies on instructor qualitative evaluations [24, 29, 17]. This opinion is often subjective and non-quantitative, but can provide good gestalt insights based on the students' online activities which makes the evaluation easier [18, 23, 13]. The same approach has also been used in software development projects in industry where the managers can view a summary of a team member's activities while evaluating their performance [31, 28, 21]. Kim et al. and Liu et al. suggested generating reports for the instructors based on version control system logs to track the students' activities and progress and intervene if needed [18, 23]. Such reports include information such as: who created the document, how many students edited the document, how many edits were made, how long the document was edited, how many words were included [18], total number of revisions, and the average number of work days [23]. Studies have shown that these types of reports can be used to track student team project progress and to intervene if necessary.

While having the instructor or team manager's opinion is a reliable method to evaluate the contributions of different team members, Lima et al. noted that managers often find this evaluation time-consuming and that is has no specific criteria for good teamwork [21]. As a result, more recent studies have focused on automatically extracting the students' share of work from a version control system [13]. For example, Ganapathy et al. evaluated group collaboration by the number of documents edited by several group members and found that better collaboration could predict a better outcome on the project [13]. El et al. similarly showed that the number of commits and the amount of lines of code added by a user are statistically significant characteristics for identifying contributions to the team.

3. DATASET

The dataset used in this study covers two consecutive fall semesters (2015 and 2016) of a CS2 Java programming course for majors. The course covers topics such as object-oriented design, testing, composition, inheritance, state machines, linear data structures, and recursion. Both course offerings were taught by the same instructors and were split into two on-campus sections. All sections included two midterm exams (referred to as Test 1 and Test 2), a final exam, lab sessions, and three projects. The first project was completed individually while the students had the option to work in pairs for the second project, and were required to do so for the third. Students were allowed to request specific teammates or

Table 1: Statistics of Each Class

Class	Java-2015	Java-2016
On-campus Students	181	206
Teaching Assistants	9	9
On-campus Instructors	2	1
Average Grade	79.7	79.9
Project 2 pairs	36	44
Project 2 selected peers	30	39
Project 2 assigned by instructor	6	5
Project 3 pairs	73	85
Project 3 selected peers	39	56
Project 3 assigned by instructor	34	29
Avg commits per repository	109	66
Max commits per repository	317	198

have them assigned by the instructor. When assigning students to teams, the instructor created balanced teams based upon similar prior performance on individual work (i.e. exam 1 and project 1). Both of the team projects included an individual component (a high-level system design and system test plan), and a team component (a system implementation). The system implementation part took about two weeks and the students were not permitted to work in a team if they failed to complete the individual task. Once students completed the individual parts and formed teams, an instructor-authored design was released and the students were required to implement it for the second stage of the project.

Students used the Eclipse IDE for the project implementation and were graded based on teaching staff and student-authored test cases, code coverage from student-authored test cases (EclEmma), coding style (SpotBugs, PMD, CheckStyle), and documentation (JavaDoc). They used Moodle as a learning management system (LMS) to access materials and Piazza as a shared discussion forum. They also used the GitHub version control system to support teamwork and track coding progress. Whenever a student made changes to their project, a difference (diff) between the currently saved version and the edited version was created showing which files had been added or removed, and which lines of code had been added or removed. Students could store these diff changes by creating a “commit”, which could serve as a checkpoint for progress. These commits were then uploaded, or “pushed”, to GitHub, along with a commit message added by the students explaining the changes that had been made.

Student projects were automatically evaluated using the Jenkins continuous integration system which monitored student GitHub repositories for changes [1]. When Jenkins detected a change, it would download the current iteration of the repository, evaluate the submissions via teaching staff test cases, and provide feedback to students via a web-based platform. Students’ grades relied both on their code passing the staff test cases as well as having enough code coverage by writing their own tests. Staff test code was hidden from students, but students could see the test numbers and topics, including hints, for any failing tests. Students were allowed to submit their code for evaluation as often as they chose. Each repository had at least one (1) commit, at most 317 commits, and on average 80 commits per repository over both semesters. We focus our analysis on students’ commit history as it reflects their coding behaviors.

As shown in Table 1, the 2015 class had 182 students and 9 teaching assistants (TAs) while the 2016 class had 206 students and 10 TAs. Both these offerings included on-campus and distant education sections but we focused on the on-campus sections for consistency.

Commit Type	Percentage	Example
Implementation	0.33	Added Constructors for inner classes
Test Cases	0.15	More test cases
Bug Fixes	0.29	Fixed logout
Documentation	0.03	Added Javadoc to the class
Style	0.04	Fixing PMD errors
Merge	0.03	Merge branch ‘master’ of ...
Other	0.11	asdf

Table 2: The distribution and an example of different commit types among manually tagged data

tency. In 2015, for the second part of projects, there were 39 pairs for Project 2 and 76 pairs for Project 3; the remaining students either failed to complete the design portion of the projects and worked alone or decided to work alone on project 2. In 2016, there were 46 pairs for Project 2, 88 pairs for Project 3, one group of three members for Project 2 and another group of three for Project 3, and the remaining students worked individually. Since the aim of our study is to understand the students’ teamwork, we focused our analysis on Projects 2 and 3 and only on teams of 2 for consistency.

4. METHODS

4.1 H1. We can automatically classify commit messages into different parts of the project.

Our dataset for 2015 contains a total of 4473 commits from Project 2 and 8224 from Project 3. For 2016 we have a total of 7432 and 10430 commits for Projects 2 and 3 respectively. Since our focus is on the students’ teamwork, we focused our analysis on commit messages of student pairs in Projects 2 and 3.

We first randomly selected and manually tagged 400 commit messages from our dataset classifying them into 7 different categories that described the commits. The tagging was done by a graduate student who had acted as a TA for this course several times before and was familiar with the structure of the projects. The categories were Implementation (I), Writing test cases (T), Bug fixing (B), Style fixing (S), Documentation (D), Merge (M), and Other (O). The distribution of these commit types among the 400 manually tagged commits and one example of each category are shown in Table 2.

In 2016, students were taught about pair-programming and were specifically asked to mention it in their commit messages. We used keyword matching of words (e.g. “pair” as well as the whole word “pp”) as some students abbreviated it to identify pair programming commits. We were able to find a total of 247 commits among all the student commits mentioning pair programming, 137 from Project 2 and 110 from Project 3, all in 2016 class.

For classifying the commit messages, we used a cascade model as shown in Figure 1. Some of these categories were easily identified by specific keywords. For example, merge commits are often auto-generated and always have the word “merge” in them, documentation commits often mention document or Javadoc keywords, style commits often mention the static analysis tools like PMD or CheckStyle, and commits that belong in none of our categories often do not have meaningful words and are easily detected using English corpus. We first removed English stop-words and lemmatized the text in the commit messages. We also added class-specific keywords to the acceptable English corpus, such as BBTP (black box test plan) or TS tests (teaching staff tests). To reduce the noise in our data and increase the accuracy of our models, we used static keyword matches to label *merge*, *documentation*, and *style* and

English corpus label *other*. For the remaining tags (i.e. Implementation, Test cases, Bugfix), we used a Binary Logistic Regression classifier for each label, using TF-IDF vector of features [33], with a maximum of 45 features and an n-gram range of 1 to 4. Each binary classifier categorized a commit message as belonging to a category or not. Similar to the previous stage, we identified commits belonging in each category and removed the already-labeled commits from the dataset for the next prediction task. Any commits remaining unlabeled in the end were labeled as the Other category. After training and testing our classifiers on our tagged sample using 5-fold cross-validation, we used the trained models to predict the commit types for the remaining unlabeled commit messages.

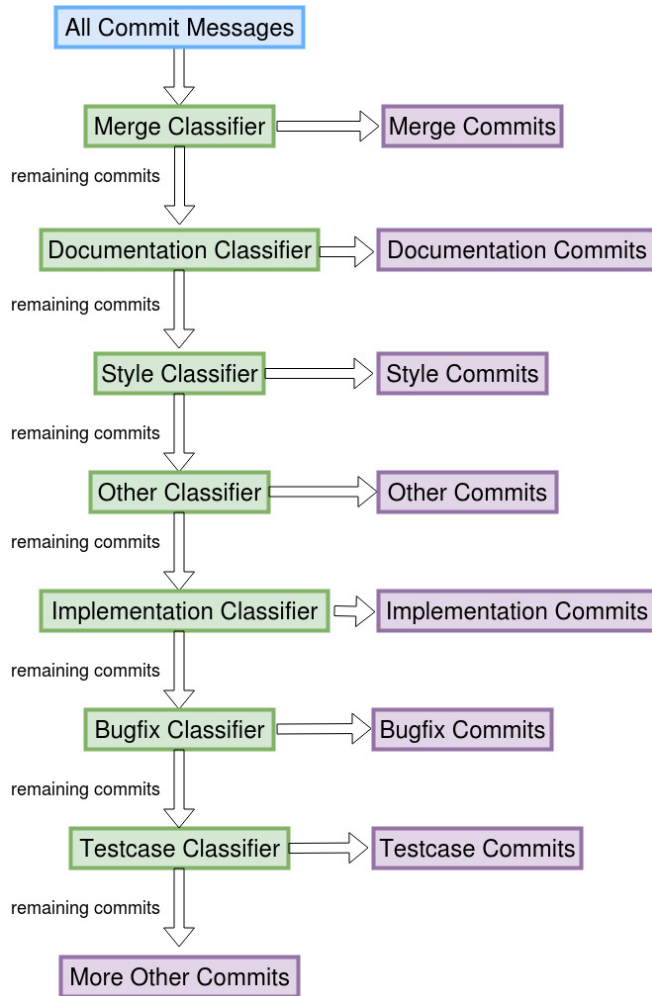


Figure 1: Cascade model for commit classification

4.2 H2. We can automatically classify student teams into Collaborative, Cooperative, or Solo-submit.

We labeled the teamwork style of the students who both worked on similar parts of the project as *collaborative* (e.g. both doing some implementation and some testing), while the teams where members both had significant contributions but mostly worked on separate parts of the project (e.g. one working on implementation and the other one on testing) were labeled *cooperative*. There were also teams where one student did the majority of work and the other student either did no work, or made small amount of changes. We labeled those teams *solo-submitting*.

To identify the teamwork styles, we randomly selected 50 repositories from each offering of the course and manually tagged them as collaborative, cooperative, and solo-submit. The tagging was done by two subject matter experts (SME), experienced TAs who are familiar with the course material and grading criteria, one of whom has acted as a TA for this course multiple times. First, a sample of 20 repositories were tagged by both SMEs with a kappa agreement of 0.88 and then the remaining repositories were tagged separately. As mentioned before, the students were able to get feedback on their code by pushing it to GitHub and checking it with the teaching staff test cases. As a result, there are many cases where the students wanted to try different fixes and submitted many commits with small changes continuously until they could pass the tests. Thus, the SMEs were asked to focus on the amount of work done by each student in each category, rather than the number of commits. To make the tagging process more consistent, we added more specific definitions for the different teamwork styles. A team where both members contributed between 30%-70% to at least two common parts of the project were considered collaborative. The teams where one member did the majority of work in some parts and the other member worked mostly on other parts were considered cooperative. If one member did not work as much, the team was labeled as solo-submit.

Identifying teamwork style by manual tagging requires a great deal of expert time and it can be difficult to come to agreement among different experts. As a consequence, for this part we focused on extracting the students' teamwork automatically by using features from their GitHub submissions, as well as their prior individual performance (exam 1 and project 1) and the way they chose their team (i.e. self-selected vs. assigned by the instructor). In discussions, one of the instructors suggested that students with prior individual grades below 60 should be considered *at-risk*. Consequently, we added new binary features reflecting the team members' risk as well. Overall, we calculated the following features for each team member, sorting the team members such that the student with fewer total added lines of code in the project would be user 0 and the student with more would be user 1 in each team.

Our final set of features included:

- The **total number of commits** for each user in the whole project as well as the number of commits in each part (Implementation, Testing, Debugging, Documentation, Merge, Style, and Other). These features can show the students' contribution as the number of commits to the whole project and to the different parts.
- The **percentage of commits** for each user in the whole project and in different parts. This feature can distinguish between 2 commits in a team with a total number of 20 commits vs. in another team with a total of 100 commits.
- The total number of **additions, deletions, files changed, and amount of change** (i.e. additions + deletions) for each user in the whole project as well as each part. Additions and deletions in GitHub are measured by the lines of code each user changes in a specific commit.
- The **average amount of additions, deletions, files changed, and amount of change** per commit for each user in the whole project as well as each part.
- The **percentage** of each students' additions, deletions, files changes, and amount of change in the whole project as well as each part. Similar to the percentage of commits, this can normalize the amount of change for each team based on their total amount of activities.
- The total and average **length of commit messages** for each

	Merge	Style	Documentation	Other
F1 score	0.98	0.99	0.99	0.95
Precision	1.00	1.00	1.00	1.00
Recall	0.96	0.98	0.99	0.90

Table 3: The performance of prediction models in finding Merge, Style, Documentation, and Other commits

user in the whole project and each part. This feature can distinguish between the members who write details about their changes and the ones who submit quick commits without much explanation.

- The total number of **pair programming commits** by each user as well as the total for the whole project. While using the total amount of pair programming is more intuitive, we believe that if all the pair programming is done on one person’s computer, it might provide some information about the dynamics of the teamwork.
- Prior individual performance for each user (i.e. exam 1 and project 1 grades). Exam 1 and project 1 take place at a similar time and before project 2 and project 3.
- Risk label ($grade < 60$). We added each student’s risk label for exam 1 and project 1 as separate features, as well as one overall risk label for the team which shows whether or not any member of the team could be considered at-risk based on exam 1 or project 1.
- The team’s selection method as a label “selected” which shows whether the students in this team requested working together or they were assigned by the instructor.

After defining and standardizing each feature, we ended up with 188 features. We used random forest feature selection as well as the recursive feature elimination (RFE) method with logistic regression to select the most important features for predicting students’ teamwork style (i.e. collaborative, cooperative, or solo-submit). Random forests in Scikit-learn library return feature importance for all the features and we can select a desired number of top features for our model [32]. The RFE method in Scikit-learn library uses the coefficients of a linear model (in our case logistic regression) to estimate feature importance and prune the least important features until reaching the desired number of features [32]. We tried different numbers of features to find the features that resulted in better F1-scores. We then used cascade binary random forest and logistic regression classifiers using the selected features to predict each project’s teamwork style. We chose these models because they are fast and they also provide us with information on what features they used and how those features contributed to the outcome, which can be useful when planning future interventions. Similar to commit classifications, these binary classifiers were trained based on belonging or not belonging to each category. We tested the accuracy of these models using 5-fold cross validation.

5. RESULTS AND DISCUSSION

5.1 H1. We can automatically classify commit messages into different parts of the project.

We first trained classifiers for the manually tagged commit messages in the categories of Style, Documentation, Merge, and Other using static matches. The F1-score, precision, and recall for these predictions are shown in Table 3.

After removing these categories, we classified the remaining tagged commits into Implementation, Tests, and Bug fixes. In the end, any commits left in no category were categorized as

	Implementation	Bug Fix	Tests	Other
F1 score	0.84	0.92	0.92	0.78
Precision	0.88	0.87	0.86	0.64
Recall	0.82	0.97	0.98	1.00

Table 4: The performance of prediction models in finding Implementation, Bug Fixes, Tests, and Other commits

	2015				2016			
	Project 2		Project 3		Project 2		Project 3	
	Count	Ratio	Count	Ratio	Count	Ratio	Count	Ratio
Implementation	848	0.25	2060	0.33	2666	0.44	4317	0.49
Test Cases	298	0.09	327	0.05	637	0.10	450	0.05
Bug Fixes	767	0.22	1433	0.23	1262	0.21	2076	0.24
Documentation	117	0.03	196	0.03	464	0.08	471	0.05
Style	141	0.04	268	0.04	457	0.08	624	0.07
Merge	367	0.11	520	0.08	173	0.03	533	0.06
Other	901	0.26	1399	0.23	433	0.07	340	0.04

Table 5: The distribution of different commit types in each year and project

Other. Since the list of Others commits changed after this stage, we calculated the accuracy of the models for this label twice, once based on the static analysis of the commit message as shown in Table 3, and another time after assigning all the commits left uncategorized to this group as shown in Table 4. The average F1-score, precision, and recall for the 5-fold cross validation for these predictive models are shown in Table 4. As shown in this table, the precision of the Other category reduced as we added the remaining uncategorized samples to this group, which means some of these samples belonged to other categories but were not found by them, but the high recall score shows that all the commits in the Other category were identified successfully.

These results support H1, showing that our prediction models are able to predict the categories of commit messages with an F1 score of 0.78 or higher. For most of the categories, the F1 score is higher than 0.9. After this step, we trained prediction models on all the tagged sample and used those models to predict the tags for the remaining untagged commit messages. The distribution of different commit messages in all the data is shown in Table 5. These distributions show us that for all the projects and all the classes, a large portion of the students’ commits belong to implementation and fixing bugs. Having very few style-based or documentation and tests commits shows that the students often fix style issues or add documentation and tests for the projects in fewer attempts. This is likely because they can check style errors and code coverage on their local platforms and submit once done, while adding features to their code and getting a functional version of the project that passes all the teaching staff test cases is often challenging and takes many attempts. Teaching staff tests were hidden from students and feedback was only available by committing code to GitHub that was then automatically executed on Jenkins. Students likely made frequent changes to address teaching staff test failures.

5.2 H2. We can automatically classify student teams into Collaborative, Cooperative, or Solo-submit.

In our SME-tagged data, we identified a total of 14 solo working teams, 55 collaborating teams and 28 cooperating teams. We removed five teams from our analysis that had more than two members or only one member contributing either because they were teams of 3 or 1 or because the members changed at some

	Total		2015		2016	
	Count	Ratio	Count	Ratio	Count	Ratio
SME tagged						
Collaborative	55	0.57	18	0.39	37	0.76
Cooperative	28	0.29	18	0.39	10	0.20
Solo-submitting	12	0.14	10	0.22	2	0.04

Table 6: Distribution of the different teamwork styles

	Logistic Regression		
	Collaborative	Cooperative	Solo-submit
F1 Score	0.61	0.67	0.84
Precision	0.51	0.70	0.90
Recall	0.78	0.64	0.78
	Random Forest		
	Collaborative	Cooperative	Solo-submit
F1 Score	0.68	0.78	0.90
Precision	0.63	0.75	0.89
Recall	0.79	0.83	0.92

Table 7: The performance of prediction models for students’ teamwork style

point. The detailed breakdown of the repositories into different styles for each year is shown in Table 6. The performance of the Random Forest classifier and the Logistic Regression with recursive feature elimination is shown in Table 7. As shown in this table, both models had similar performance in predicting the students’ teamwork style, Random Forest performed slightly better, with solo-submit being the easiest to predict and collaborative being the most difficult.

The random forest model worked best with 12 features and the logistic regression worked best with 26. Since random forest performed better at predicting the teamwork style, we analyzed the top features for these random forest models. As these features show, the students’ activities in different parts of the project and their prior individual performance were good predictors for their teamwork. Most of the top 12 features selected by random forest for the collaborative, cooperative, and solo-submit classifiers were specific to each of the teamwork style, but some of the features like the *Average deletion per commit for user 0* were common across styles. The *Percentage of commits for the whole project* for both users were the top features for predicting Solo-submitting, while the *Percentage of commits* in different categories and the *Students’ prior performance* were more predictive for Collaborative and Cooperative. Surprisingly, the *Number of pair-programming activities* were not among the top features, which might be because the students do not always record pair programming in their commit messages.

These prediction models show us that we can identify the students’ teamwork style, especially solo-submitting by using automatically generated features from their commit history and their contributions to the different parts of the project. One might assume that looking at the repositories and the students’ number of commits should be sufficient for identifying solo-submitters. However, as the SMEs noticed, deciding whether both members had significant contributions to the team was challenging and time-consuming, even for experienced TAs. Most of the defined metrics such as the *Amount of implementation commits* or the *Percentage of commits* in a repository can be extracted automatically early in the semester. As a result, using predictive models with these features could help identify the need for early

intervention, for example when teamwork habits indicate that solo-submit may eventually happen in a team.

6. LIMITATIONS AND FUTURE WORK

There are three main limitations to our work. First, our dataset is drawn from a single course. Thus it is possible that the observed results will not generalize to courses with a different team structure or grade breakdown. We do argue however that the analytical methods we chose are general and we plan to evaluate them on different courses in a future study. Second, our classification of the student teams was based solely on their observable online behavior and did not consider offline activities. It is possible that offline behaviors such as students meeting face to face, or exchanging code through other media, might affect our results.

7. CONCLUSION

In this study, we first hypothesized that we could automatically identify students’ activities on different parts of development projects based on the text of their commit messages. We later hypothesized in H2 that we could automatically identify different teamwork styles among students using their online submissions and which parts they belong to. For the first part, we manually tagged 400 commit messages as belonging to different parts of the projects as Implementation, Testing, Debug, Style, Documentation, Merge, and Other. We then used TF-IDF features and a logistic regression to automatically label the remaining commits. To analyze different styles in students’ teamwork, we manually labeled 100 GitHub repositories of student projects in two offerings of a Java introductory course for CS majors as “Collaborative”, “Cooperative”, or “Solo-submit”. We then used several measures based on the students’ activities on GitHub, their prior performance, and whether they chose their teammate to automatically label all the student repositories in these classes. We observed that these models were able to achieve an F1 score of 0.68 or better for different categories, which supported our hypothesis that students’ online activities can identify their teamwork style.

The students in these classes were not graded for their amount of contributions on GitHub. As a result, students were able to split the work among themselves based on their choices and what we observed here was their natural behaviors. This makes the findings in this study more likely to apply to other classes since the students’ teamwork styles were not directed by the course structure.

The findings of this study can be used to analyze which styles of teamwork lead to better performance in classes. Eventually, the findings can help design adaptive support platforms for the instructors to observe a summary of the students’ activities and possible red flags in their behavior such as solo-submitting. The instructors can then plan interventions in a timely manner to help the students to better engage with authentic team projects in the class.

8. ACKNOWLEDGEMENTS

This research was supported by NSF 1821475 “Concert: Coordinating Educational Interactions for Student Engagement” Collin F. Lynch, Tiffany Barnes, and Sarah Heckman (Co-PIs).

9. REFERENCES

- [1] Jenkins. <https://jenkins.io/>.
- [2] M. Ahmadzadeh, D. Elliman, and C. Higgins. An analysis of patterns of debugging among novice computer science students. In *Acm sigcse bulletin*, volume 37, pages 84–88. ACM, 2005.
- [3] A. Bandura and R. H. Walters. *Social learning theory*, volume 1. Prentice-hall Englewood Cliffs, NJ, 1977.
- [4] T. F. Barr, A. L. Dixon, and J. B. Gassenheimer. Exploring the “lone wolf” phenomenon in student teams. *Journal of Marketing Education*, 27(1):81–90, 2005.
- [5] P. Blikstein. Using learning analytics to assess students’ behavior in open-ended programming tasks. In *Proceedings of the 1st international conference on learning analytics and knowledge*, pages 110–116. ACM, 2011.
- [6] A. S. Carter, C. D. Hundhausen, and O. Adesope. The normalized programming state model: Predicting student performance in computing courses based on programming behavior. In *Proceedings of the eleventh annual International Conference on Computing Education Research*, pages 141–150. ACM, 2015.
- [7] A. S. Carter, C. D. Hundhausen, and O. Adesope. Blending measures of programming and social behavior into predictive models of student achievement in early computing courses. *ACM Trans. Comput. Educ.*, 17(3), Aug. 2017.
- [8] P.-Y. Chao. Exploring students’ computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Education*, 95:202–215, 2016.
- [9] C. Chen, Y. Hong, and P. Chen. Effects of the meetings-flow approach on quality teamwork in the training of software capstone projects. *IEEE Transactions on Education*, 57(3):201–208, Aug 2014.
- [10] J. Chen, G. Qiu, L. Yuan, L. Zhang, and G. Lu. Assessing teamwork performance in software engineering education: A case in a software engineering undergraduate course. In *2011 18th Asia-Pacific Software Engineering Conference*, pages 17–24. IEEE, 2011.
- [11] I. D. Coman, P. N. Robillard, A. Sillitti, and G. Succi. Cooperation, collaboration and pair-programming: Field studies on backup behavior. *Journal of Systems and Software*, 91:124–134, 2014.
- [12] S. B. Feichtner and E. A. Davis. Why some groups fail: A survey of students’ experiences with learning groups. *Organizational Behavior Teaching Review*, 9(4):58–73, 1984.
- [13] C. Ganapathy, E. Shaw, and J. Kim. Assessing collaborative undergraduate student wikis and svn with technology-based instrumentation: Relating participation patterns to learning. In *Proceedings of the American Society of Engineering Education Conference*. Citeseer, 2011.
- [14] L. Glassy. Using version control to observe student software development processes. *Journal of Computing Sciences in Colleges*, 21(3):99–106, 2006.
- [15] M. Hoegl and H. G. Gemuenden. Teamwork quality and the success of innovative projects: A theoretical concept and empirical evidence. *Organization science*, 12(4):435–449, 2001.
- [16] R. Hosseini, A. Vihavainen, and P. Brusilovsky. Exploring problem solving paths in a java programming course. *Proceedings of the 25th Workshop of the Psychology of Programming Interest Group*, 2014.
- [17] P. Imbrie, J. C. Immekus, and S. J. Maller. Work in progress-a model to evaluate team effectiveness. In *Proceedings Frontiers in Education 35th Annual Conference*, pages T4F–12. IEEE, 2005.
- [18] J. Kim, E. Shaw, H. Xu, and G. Adarsh. Assisting instructional assessment of undergraduate collaborative wiki and svn activities. *International Educational Data Mining Society*, 2012.
- [19] H.-J. Lee. Peer evaluation in blended team project-based learning; what do students find important? In *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, pages 2838–2842. Association for the Advancement of Computing in Education (AACE), 2009.
- [20] H.-J. Lee, H. Kim, and H. Byun. Are high achievers successful in collaborative learning? an explorative study of college students’ learning approaches in team project-based learning. *Innovations in Education and Teaching International*, 54(5):418–427, 2017.
- [21] J. Lima, C. Treude, F. F. Filho, and U. Kulesza. Assessing developer contribution with repository mining-based metrics. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 536–540, Sep. 2015.
- [22] Y.-T. Lin, C.-C. Wu, T.-Y. Hou, Y.-C. Lin, F.-Y. Yang, and C.-H. Chang. Tracking students’ cognitive processes during program debugging-an eye-movement approach. *IEEE transactions on education*, 59(3):175–186, 2016.
- [23] Y. Liu, E. Stroulia, K. Wong, and D. German. Using cvs historical information to understand how students develop software. In *Proceedings of the International Workshop on Mining Software Repositories, Edinburgh, Scotland. IET*, 2004.
- [24] J. B. Main and M. Sanchez-Pena. Student evaluations of team members: Is there gender bias? In *2015 IEEE Frontiers in Education Conference (FIE)*, pages 1–6. IEEE, 2015.
- [25] K. Mierle, K. Laven, S. Roweis, and G. Wilson. Mining student cvs repositories for performance indicators. In *ACM SIGSOFT Software Engineering Notes*, volume 30, pages 1–5. ACM, 2005.
- [26] C. Murphy, G. Kaiser, K. Loveland, and S. Hasan. Retina: helping students and instructors based on observed programming activities. *ACM SIGCSE Bulletin*, 41(1):178–182, 2009.
- [27] P. Näykki, S. Järvelä, P. A. Kirschner, and H. Järvenoja. Socio-emotional conflict in collaborative learning-a process-oriented case study in a higher education context. *International Journal of Educational Research*, 68:1–14, 2014.
- [28] T. Nguyen and C. Chua. Predictive tool for software team performance. In *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*, pages 373–376. IEEE, 2016.
- [29] S. G. Northrup and D. A. Northrup. Multidisciplinary teamwork assessment: Individual contributions and interdisciplinary interaction. In *Proceedings. Frontiers in Education. 36th Annual Conference*, pages 15–20. IEEE, 2006.
- [30] B. Oakley, R. M. Felder, R. Brent, and I. Elhaji. Turning student groups into effective teams. *Journal of student centered learning*, 2(1):9–34, 2004.
- [31] R. M. Parizi, P. Spoletini, and A. Singh. Measuring team members’ contributions in software engineering projects using git-driven technology. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–5, Oct 2018.
- [32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [33] J. Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. Piscataway, NJ, 2003.
- [34] K. L. Reid and G. V. Wilson. Learning by doing: introducing

- version control as a way to manage student assignments. In *Acm Sigcse Bulletin*, volume 37, pages 272–276. ACM, 2005.
- [35] G. Salomon and T. Globerson. When teams do not function the way they ought to. *International journal of Educational research*, 13(1):89–99, 1989.
 - [36] A. Seers. Team-member exchange quality: A new construct for role-making research. *Organizational Behavior and Human Decision Processes*, 43(1):118–135, 1989.
 - [37] J. Spacco, P. Denny, B. Richards, D. Babcock, D. Hovemeyer, J. Moscola, and R. Duvall. Analyzing student work patterns using programming exercise data. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 18–23. ACM, 2015.
 - [38] S. Uchida, A. Monden, H. Iida, K.-i. Matsumoto, and H. Kudo. A multiple-view analysis model of debugging processes. In *Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium on*, pages 139–147. IEEE, 2002.
 - [39] P. Van den Bossche, W. H. Gijselaers, M. Segers, and P. A. Kirschner. Social and cognitive factors driving teamwork in collaborative learning environments: Team learning beliefs and behaviors. *Small group research*, 37(5):490–521, 2006.
 - [40] L. van der Duim, J. Andersson, and M. Sinnema. Good practices for educational software engineering projects. In *29th International Conference on Software Engineering (ICSE’07)*, pages 698–707, May 2007.
 - [41] A. Vihavainen, M. Luukkainen, and J. Kurhila. Using students’ programming behavior to predict success in an introductory mathematics course. In *Educational Data Mining 2013*, 2013.
 - [42] C. Watson, F. W. Li, and J. L. Godwin. No tests required: comparing traditional and dynamic predictors of programming success. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 469–474. ACM, 2014.
 - [43] M. Wen, K. Maki, S. Dow, J. D. Herbsleb, and C. Rose. Supporting virtual team formation through community-wide deliberation. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):109, 2017.

Process based Analysis on Scientific Inquiry Tasks Using Large-scale National Assessment Dataset

Tao Gong, Lan Shuai,
Burcu Arslan, Yang Jiang
Educational Testing Service
tgong@ets.org

ABSTRACT

This paper investigates differences in students having various scores when designing controlled experiments in two types of scientific inquiry tasks (a fair test and an exhaustive test). We measure temporal features of preparation time and execution time, which reflect respectively the process of question understanding and answer planning and that of executing the control-of-variables strategy in answer formulation. We also measure mean execution time per answering event to reflect the efficiency of answering events. Results show that: in the fair test, the full score students showed less execution time than the lowest score ones; in the exhaustive test, the full score students showed more execution time than the lowest score ones; but in both tests, the high-performing students had less mean execution time than the low-performing ones. These results reveal that despite test differences, students who appropriately apply the control-of-variables strategy in these tests are more goal-directed and efficient in planning and executing response strategies than those who fail to do so. This study provides process-based features and large-scale evidence of scientific inquiry practice in educational assessment.

Keywords

Control-of-variables strategy, preparation time, execution time

1. INTRODUCTION

Scientific inquiry refers to the activities by which students develop knowledge of scientific ideas and understand how to investigate the natural world in a scientific way [1]. In STEM education, scientific inquiry skills have been emphasized as a key goal of scientific literacy [2,3], and scientists and science educators have advocated teaching science as inquiry [4–8]. Among scientific inquiry activities (see [6] for overview), *planning*, *designing*, and *carrying out investigations* have long become a principal focus of children's and youngsters' scientific inquiry practices [9,10]. Many studies aim to investigate, based primarily on response data, how students design controlled experiments by constructing related conditions for comparison.

Fair tests and exhaustive tests have been widely adopted to examine how students plan, design, and carry out controlled experiments. A *fair test* (see an example in Sec. 2.2) refers to a controlled investigation carried out to answer a scientific question

about the effect of a target variable. To control for confounding factors and be scientifically sound, students are supposed to apply a *control-of-variables strategy* (CVS) [9,11] to ensure that: (a) all the other variable(s) are kept constant; and (b) only the variable(s) under investigation is changed across conditions for comparison. Only in such a fair setting, the effect of the target variable(s) can be explicitly observed, since the other variables remain constant across conditions. Students can complete the task by choosing, among a large number of possible combinations of variables, one or a few conditions that meet the fair test requirement.

An *exhaustive test* (a.k.a. combinatorial test, [12,13]) (see an example in Sec. 2.3) requires constructing, physically or mentally, all possible combinations of given variables to address inquiry on which conditions could cause a specific outcome. Like fair tests, students in exhaustive tests also need to control target variables to construct combinations, but the number of possible combinations is generally smaller than that in fair tests. In exhaustive tests, students are asked to enumerate all combinations; in fair tests, students only need to select one (or a few) condition that meet the requirement. In this sense, exhaustive tests require more cognitive resources especially in situations with not easily foreseen combinations. How to conduct an exhaustive test is taught and learned late in science education, and items assessing such skill often lie in the 8th, 12th, or higher-grade assessments [3].

CVS is required in both types of tests. Among other types of procedural knowledge, or “process skills”, CVS is deemed central to early science instruction [14]. Existing research shows that children, adolescents, and adults with low scientific inquiry expertise tend to have difficulty in applying CVS [9,10,15]. However, due to lacking measures on processes of scientific inquiry, existing studies focus primarily on students' responses.

In modern digitally-based assessment programs (e.g., National Assessment of Educational Progress (NAEP)), technology-enhanced (TE) items have been used to study scientific inquiry practice. The interactive nature of such items allows recording not only final submitted answers, but also the process whereby students formulate their answers via a series of drag-and-drop, (de)selection, or correction actions. Obtained process data can gather additional evidence on what students do during inquiry [16–18]. TE items have now touched upon many disciplines, including math, science, and social science [18–21], and process data obtained have covered not only observable behaviors of test-takers in problem solving but also frequencies and durations of such actions, both contributing to illustrating the mastery phases in scientific inquiry and response strategies of students [22–25]. In addition, process-based analyses help discover the aspects where students of different scores differ, and lead to better understanding of the cognitive framework of scientific inquiry.

Tao Gong, Lan Shuai, Burcu Arslan and Yang Jiang "Using Process Data to Evaluate Scientific Inquiry Practice in Technology-Enhanced Assessment" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 417 - 423

Rather than concrete events recorded in obtained process data of TE items, the time needed for different stages of scientific inquiry has been undervalued in recent research of scientific inquiry or problem solving [26]. Temporal information can reveal different stages of problem solving, clarify performance patterns of students with different levels of problem solving competency, and allow inferring something about the cognitive processes occurring at various phases of problem solving.

Noting these, this study aims to investigate the scientific inquiry practice, to be specific, the practice of designing controlled experiments by applying the CVS in fair and exhaustive tests. By evaluating relations between defined process-based, temporal measures and students' performance gauged by scores, we aim to address the following two research questions:

- (a) What are the process-based characteristics of the high-/low-performing (indicated by different scores) students in the tests?
- (b) Are these process-based characteristics consistent across the fair and exhaustive tests?

Answers to these questions can benefit the general discussions on scientific inquiry practice, especially whether the CSV strategy manifest differently across various types of inquiry tasks. They also provide actionable feedback to teaching and learning the skills required in scientific inquiry tasks. Moreover, this study enriches the literature of using process data and derived features to address theoretical issues in educational assessment.

In the rest of the paper, we describe the NAEP science fair test and exhaustive test used in this study, define the process-based measures, and describe the analysis plan. Then, we report the results, discuss the research questions accordingly, and conclude the study by highlighting theoretical or operational applications of process-based analyses in education and psychology research.

2. METHODS AND MATERIALS

2.1 NAEP Science Tasks

Our study uses the 2018 NAEP science pilot tasks. NAEP is a congressionally mandated, nationwide digital assessment project administered by the National Center for Education Statistics (NCES) in the Institute of Education Sciences of the U.S. Department of Education. NAEP provides large-scale, regular assessments on many disciplines (e.g., math, reading, writing, science, etc.). All the assessments are designed and updated by content specialists, education experts, and teachers from around the U.S. Participants of the tests are grades 4 (~9-year-olds), 8 (~13-year-olds) and 12 (~17-year-olds) students. Along with the assessment, survey data of students, teachers, and schools are gathered, covering students' demographical information (gender and ethnicity), special programs, self-evaluation of performance, etc. NAEP has now become one of the largest and most important national assessments of what U.S. students know and can do.

The 2018 assessment was conducted by the NAEP field staff, who went into schools across the nation to administer tasks on students from the NAEP sample. The science tasks were administered on NAEP-provided tablets with an attached keyboard and earbuds. Students had 60 minutes to complete the questions in the given task. Tutorials and surveys were given throughout the test.

A total of 32 science tasks were designed for the 2018 NAEP pilot test, some of which were administered on grades 4, 8, and 12 students. Our study focuses on a fair test and an exhaustive test,

which were administered respectively on grade 8 and 12 students. This choice was due to three considerations. First, lower grade students have not been taught how to solve both types of tests, so we avoid tasks administered on grade 4 students. Second, since fair tests were administered mostly on grade 4 and 8 students but exhaustive tests were administered mostly on grade 12 students, we could not select fair tests and exhaustive tests administered on students of the same grade. Third, to properly answer the two chosen tests, students needed to submit similar numbers of distinct answers, which avoided possible interference from cognitive load in students' answer formulation process.

Due to the privacy and secure nature of the NAEP data, we use conceptually equivalent tasks (*cover tasks*) to disguise the content and context of the real tasks. Cover tasks have similar underlying structures and require similar cognitive processes to solve.

2.2 Fair Test, Scoring Rubric, Students

This test came from an earth and space science task. Its cover test is as follows (see Figure 1). A city near a mountain suffers from north winds each year. Its government plans to test the wind-blocking power of three types of trees, which can be planted at the north side of the mountain. After simple instructions of the task, in the fair test scene of the task, students are asked to drag each type of trees and drop them at one of the four virtual mountains resembling the real one near the city. Students can drop the trees at the foot (low), side (medium), or peak (high) of the north side of a mountain. Each mountain holds at most one type of trees, and each type can only be planted at one mount. Students can move trees from one position/mountain to another. After selecting the locations of the three types of trees, students can click on the on-screen "Submit" button to trigger the experiment, and the wind speeds before and after passing over the mountains are shown on the screen. By default, one mountain is left without any tree.

There are two types of variables in this fair test: tree type and tree position on mountain. To illustrate the effect of trees, students must control the positions of the trees to be identical across conditions (mountains). There are in principle $3 \times 3 \times 3 \times P(4,3) = 648$ choices for students to plant the trees, among which $3 \times P(4,3) = 72$ choices meet the fair test requirement.

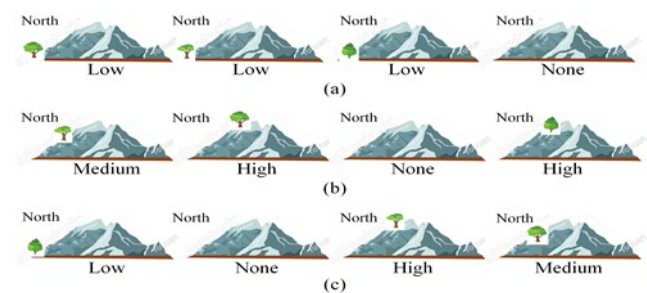


Figure 1. Example answers in the fair test. "Low", "Medium", "High" denote positions (foot, side, peak) of trees in the north side of the mountain. "None" means no tree planted. In (a), the first "Low" indicates that one type of trees are planted at the foot of the mountain, the second and third "Low" indicate that the other types of trees are planted on the second and third mountains, and "None" means no trees on the fourth mountain. The scoring rubric ignores tree types and the mountain without trees, the submitted answer can thus be denoted by the positions of trees in three mountains.

Table 1 shows the scoring rubric of this test. The rubric ignores tree type, since students cannot put the same type of trees in two mountains or two positions of one mountain. It also ignores the mountain without trees (“None”), since this is a default condition of the test; no matter how to answer the test, one mountain must be left without trees. A complete comparison to show the effects of trees requires the condition without trees, but in this test, students are not required to set up this condition. The rubric focuses on the target variable of tree positions across mountains. Answers meeting the fair test requirement receive a full score (3), those partially meeting the requirement get a partial score (2), and those not meeting the requirement have the lowest score (1).

Table 1. Scoring rubric of the fair test.

Score	Rubric
3	Choices of trees have the same positions on three mountains (e.g., Low; Low; Low in Figure 1(a))
2	Two types of trees are on the same positions of mountains (e.g., Medium; High; High in Figure 1(b))
1	Positions of the three types of trees on mountains are all distinct (e.g., Low; High; Medium in Figure 1(c))

This task was administered to 1,657 (825 females) grade 8 students. The response and process data of 1,607 (800 females) students were recorded in the fair test for analyses. Fifty-one students, due to various reasons, quit before reaching the fair test.

2.3 Exhaustive Test, Scoring Rubric, Students

This test came from a life science task. Its cover test is as follows. Farmers attempt to cultivate flowers with a special color in a natural way (without using any fertilizers) or using two types of fertilizers. After simple instructions of the task, students are asked to design an experiment to show which way has the highest probability to induce the target color. They can set up a condition by selecting (or not) any (or both) type of the fertilizers. After setting up a condition, they can click on the on-screen “Save” button to save the condition. They can also click on a saved condition and click on the “Delete” button to remove it. After setting up and saving many conditions, students can click on the “Submit” button to submit saved conditions as final answers.

This is a typical exhaustive test with four possible combinations of the variables (see Figure 2). The conditions no fertilizer (Figure 2(a)) and both fertilizers (Figure 2(d)) are not easily foreseen.

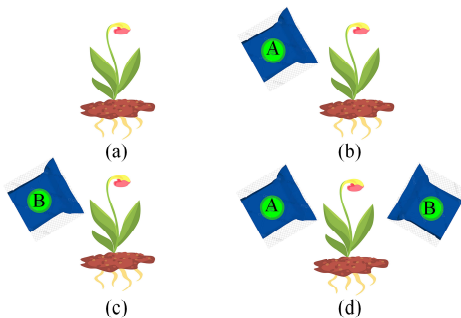


Figure 2. All combinations in the exhaustive test: (a) None; (b): A; (c): B; (d): A + B.

Table 2 shows the scoring rubric of the test. It has four scales, among which partially high (3) and partially low (2) are classified by submitted saved conditions, especially whether they include

the not-easily foreseen conditions. Whether the rubric reasonably classifies students’ skill levels is not the focus of this paper.

This task was administered to 2,869 (1,360 females) grade 12 students. The response and process data of 2,726 (1,285 females) students were recorded in the exhaustive test for the analyses. Due to various reasons (e.g., early quitting or glitches in data capture), the process data of 173 students were missing.

Table 2. Scoring rubric of the exhaustive test.

Score	Rubric
4	Saved conditions cover all four conditions in Figure 2
3	Saved conditions do not include the condition of Figure 2(a), OR do not include the condition of Figure 2(b) or Figure 2(c)
2	Saved conditions do not include the condition of Figure 2(d), OR do not include the conditions of Figure 2(b) or Figure 2(c), OR do not include both the conditions of Figure 2(a) and Figure 2(d)
1	Saved conditions do not match the above cases

2.4 Process-Based Measures

The NAEP digital assessment system can record students’ process data in these interactive TE items. Such data consisted of a list of activity logs plus their time stamps. Activities included user events (e.g., drag-and-drop, save, delete, or correct, etc.) and system events (e.g., play instructions or video clips). They allow reconstructing submitted answers, tracing sequences of students’ drag-and-drop or saving/deletion/correction actions, and durations of these activities. Based on such data, we propose and measure three temporal measures, namely preparation time, execution time, and mean execution time per answering event.

Preparation time (PT) is defined as the duration between students enter the test scene and make their first answer-related event, such as drag-and-drop one type of trees, select a fertilizer, or save a condition without any fertilizers. Before the test scenes, students were given instructions and practice trials on how to set up answers in the test scenes. Therefore, PT does not involve the time students spent on getting familiar with the system. PT reflects the time for students to read and understand instructions, as well as think and get ready to formulate their answers.

Execution time (ET) is defined as the duration between students’ first and last answer-related events. The ending time point of ET was not when students clicked on the submission button. This is because we do not know exactly whether students reviewed their answers after making their last drag-and-drop or selection event before submission. If they did review and made corrections, the measure can certainly capture such reviewing event; if they did not make any changes, it is unclear whether the time between the last answer-related event and the submission event was spent on reviewing. Many students actually clicked on the “Submit” button immediately after the last answer-related event.

ET is the sum of the durations of different numbers of answer related events. In the fair test, such events include dragging and dropping a type of trees to a mountain or moving one from one mountain to another; in the exhaustive test, such events include selecting one or two fertilizers, or saving a condition or cancelling a saved one. Students having different performances may put different efforts when conducting these events, and different tasks may require different numbers of events to formulate answers, which already lead to different ET. Noting these, we also calculate

the *mean execution time per answering event* (MET). MET is operationalized as the execution time divided by the number of answering events. ET reflects the total efforts made by students to construct answers, including setting up, revising or (possibly) reviewing their choices, whereas MET reflects the average effort made to construct their answers, and it controls the effect induced by different numbers of events.

Apart from temporal measures, one can also measure the numbers of answer related events made by students during the answering process. However, for students who conducted the same number of answering events, this count-based measure cannot clarify how much effort each event costs to these students; more events may not always require more efforts, since an efficient test-taker can conduct many events in a short period of time; and more events alone cannot predict performance in different tests, since some of the events could be answer revisions, which simply indicate low efficiency. The temporal measures defined in our study avoid these confusions and are more informative of students' degrees of efficiency in designing controlled experiments in those tasks.

2.5 Analyses

For each dataset, we take a 98% winsorization estimation [27] to remove spurious outliers. We also remove the missing values.

We conduct two types of analyses. First, we check how many students appropriately applied the required CVS in the tests based on score distributions and illustrate the frequent (top 10) correct or incorrect submitted answers. Second, treating score as a ranked variable, we conduct the Kruskal-Wallis test [28], a non-parametric version of ANOVA test, to compare students' scores and the three measures across score groups. If the omnibus test produces a significant p -value, we conduct the Wilcoxon signed-rank test on pair-wised score groups to clarify which two groups have different population means of the measures. This test is also non-parametric. These two statistical tests provide direct evidence on the relation between students' performance (scores) and the process-based measures. The tests are implemented using the `kruskal.test` and `wilcox.test` functions in the *stats* package in R 3.6.1 [29]. For each task, there are three Kruskal-Wallis tests respectively on three measures, accordingly, the critical p value for identifying significance is set to $.05/3 \approx .0167$.

3. RESULTS

3.1 Fair Test

In this test, 41.4% of the students had the lowest score (1), and only 29.5% properly applied the CVS and got the full score (3). The rest (29.1%) received a partial score (2).

Figure 3 shows the top 10 frequent answers submitted by students. It shows that "Low; Low; Low" is the most frequent correct answer, but other correct answers like "Medium; Medium; Medium" and "High; High; High" are less so. In addition, "Low; Medium; High" is the most common wrong answer. Its variations, such as "High; Medium; Low" or "Low; High; Medium", are also frequent, but all of them receive the lowest score (1) (see Table 1). Answers receiving a partial score (2) (e.g., "Medium; Low; Medium") are less frequent, compared to other types of answers. These results indicate that over 70% of students did not properly apply the CVS strategy in this scientific inquiry task.

Table 3 shows the means and standard errors of the process-based measures in each score group. The Kruskal-Wallis tests report significant differences in PT ($\chi^2 = 12.2$, $df = 2$, $p = .002$), ET ($\chi^2 =$

89.916, $df = 2$, $p < .001$), and MET ($\chi^2 = 64.776$, $df = 2$, $p < .001$) between score groups. Table 4 shows the Wilcoxon signed-rank tests results. It reveals that the full score students had significantly shorter PTs, ETs, and METs than the lowest and partial score students, but these measures were not significantly different between the lowest and partial score students.

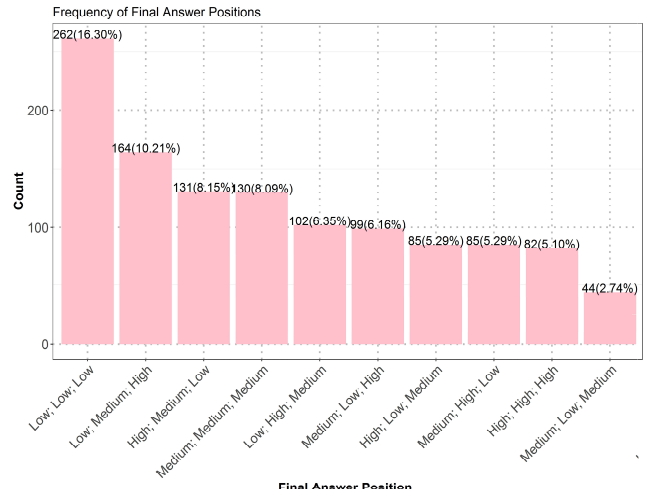


Figure 3. Top 10 frequent answers in the fair test. Numbers on top of bars are numbers of students and those inside brackets are proportions of students.

Table 3. PT, ET and MET across score groups. Numbers (in seconds) outside brackets are means and those inside are standard errors.

Score	PT	ET	MET
1	85.571 (1.166)	41.330 (1.098)	5.125 (.091)
2	85.154 (1.407)	38.807 (1.216)	4.958 (.103)
3	79.745 (1.172)	29.082 (1.081)	4.138 (.090)

Table 4. Wilcoxon signed-rank test results in the fair test. "1" to "3" in the first column denote score groups. Values outside brackets are test statistics, and those inside are p values. Significant results are marked in bold.

	PT	ET	MET
1v2	158942 (.527)	163023 (.016)	158766 (.548)
1v3	176639 (.001)	2038350.5 (.001)	199945.5 (.001)
2v3	120966.5 (.014)	139637.5 (.001)	136592.5 (.001)

3.2 Exhaustive Test

In this test, 25.2% of the students received the lowest score (1), and 33.9% properly applied the CVS strategy and received the full score (4). The rest received the partially high (3) (34.1%) and partially low (2) (6.8%) scores.

Figure 4 shows the top 10 frequent answers, among which "A; B; A + B; None" and its variations "A; A + B; B; None" and "A + B; A; B; None" receive the full score (4), but they are not frequent compared to the answers "A + B", "B", "A", and "None", which are among the most frequent answers and receive the lowest score (1) (see Table 2). The answers having partially high (e.g., "A; A + B; None") or low (e.g., "A; A + B") scores are less frequent. These results show that many students did not have the required scientific inquiry skill.

Table 5 shows the means and standard errors of the process-based measures in each score group. The Kruskal-Wallis test report significant differences in PT ($\chi^2 = 127.69$, $df = 3$, $p < .001$), ET ($\chi^2 = 332.88$, $df = 3$, $p < .001$), and MET ($\chi^2 = 238.93$, $df = 3$, $p < .001$) between the score groups. Table 6 shows the Wilcoxon signed-rank tests results. It reveals that the lowest score students had significantly longer PTs than the students from other score groups. Unlike the fair tests, the lowest score students had significantly shorter ETs than the full and partial score students. Like the fair tests, the lowest score students had significantly longer METs than the full score students.

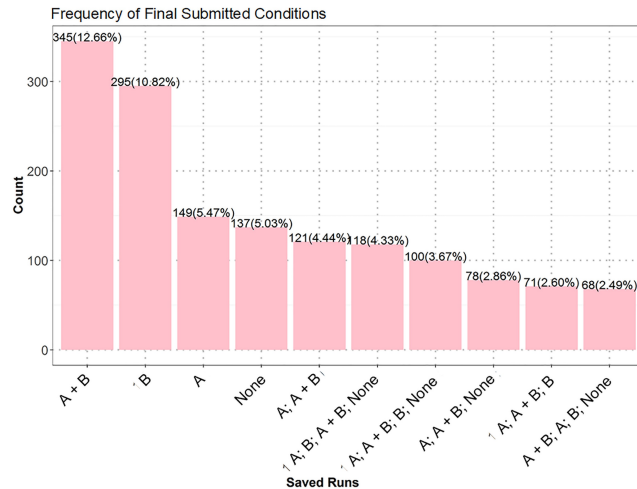


Figure 4. Top 10 frequent answers in the exhaustive test. Numbers on top of bars are numbers of students and those inside brackets are proportions of students.

Table 5. PT, ET, and MET across score groups. Numbers (in seconds) outside brackets are means and those inside are standard errors.

Score	PT	ET	MET
1	9.056 (.325)	24.949 (.922)	5.502 (.144)
2	6.797 (.439)	41.623 (1.804)	3.520 (.113)
3	7.105 (.207)	31.700 (.715)	3.899 (.070)
4	5.714 (.172)	42.523 (.763)	3.140 (.051)

Table 6. Wilcoxon signed-rank test results in the exhaustive test. “1” to “4” in the first column denote score groups. Values outside brackets are the test statistics, and those inside are p values. Significant results are marked in bold.

	PT	ET	MET
1v2	75018.0 (<.001)	29065 (<.001)	83948 (<.001)
1v3	372475.5 (<.001)	215673.5 (<.001)	400288.5 (<.001)
1v4	422941.5 (<.001)	128978.5 (<.001)	458813.5 (<.001)
2v3	84443.5 (.693)	11656 (<.001)	80219.5 (.147)
2v4	98433.5 (<.005)	81207 (.284)	100851 (<.001)
3v4	501936.5 (<.001)	274362.5 (<.001)	531023.5 (<.001)

4. DISCUSSIONS

Based on two NAEP science tasks (a fair test and an exhaustive test) and three process-based temporal features, we dig out, from both response and process data, the differences and similarities between the high-/low-performing students in those two typical types of scientific inquiry practice.

As for response, the score distributions illustrate that many (over 70%) grade 8 or 12 students failed to properly apply the control-of-variables strategy in the fair and exhaustive tests, consistent with the previous literature [9]. In addition, in the fair test (see Figure 3), the most commonly wrong strategy is to vary both variables’ levels at the same time, e.g., “Low; Medium; High” and its variations. This is also shown in previous observations [17]. In the exhaustive test (see Figure 4), the most commonly wrong strategy is to save only one of the four possible conditions as in Figure 2. This indicates that the low-performing students probably did not have the intention or the capability to design a controlled experiment but simply guessed an answer.

As for process, rather than specific actions or sequences of drag-and-drop actions as in recent studies on TE items [30], our study defines temporal features and adopts non-parametric statistical tests on these stage-level features to reveal quantitative differences between the high- and low-performing students.

The statistical tests collectively show that: in terms of preparation, the full score students spent less time before making their first answering related activity in both the fair and exhaustive tests, which are consistent with other studies [30]. Longer preparation time in the lowest score students shows that such low-performing students might have difficulty in quickly grasping the instructions or need more time to think before taking any action, whereas the high-performing students could efficiently grasp the instructions and foresee the required conditions. These results suggest that the different performances between the full and lowest score students have already manifested at the early stage of scientific inquiry practice, where no answer is formulated. In other words, whether a student can appropriately apply the control-of-variable strategy in a fair task could be highly correlated with whether he or she can efficiently grasp the instruction at the beginning of the task.

In terms of execution time, there exist differences between the fair and exhaustive tests. In the fair test, the lowest score students spent longer time on conducting the drag-and-drop actions to construct answers. As shown in Figure 3, their submitted answers after such a long execution time still failed to meet the fair test requirements. This echoes the fact that these students did not follow the instructions nor get well prepared for the fair tests. To be specific, in the fair test, the minimum number of events to construct an answer was three (dragging and dropping each type of trees respectively to the same or different locations of three mountains). Two possible situations lead to longer execution time in the lowest score students: they conducted many revisions to their early choices, or spent more time on conducting each activity, indicating their hesitation or uncertainty about their choices, or more time needed to come up with a solution due to a lack of relevant domain knowledge. Here, the results of mean execution time per answering event (see Table 4) reveal that no matter how many revisions they conducted, on average, the lowest score students spent more time on setting up each of their answers than the full score students; i.e., the full score students were more efficient than others.

In the exhaustive test, constructing all possible conditions is not trivial and requires more resources and related events. As shown in Table 5, the lowest score students spent shorter time in constructing or revising their saved conditions, whereas the full score students spent longer time in doing so. As in Figure 4, the lowest score students (and those having partial scores) did not save enough conditions, but the full score students submitted each

of the possible conditions as required by the test. Therefore, the longer execution time of the full score students reflects the fact that these high-performing students had endeavored to set up all required conditions before the final submission. By contrast, the shorter execution time of the lowest score (and partial score) students indicates that: (a) these low-performing students did not spend much time on exploring possible conditions but completed the test by submitting lack-of-thinking results, indicating their low motivation or lack of engagement in problem solving; or (b) throughout the test, they might not realize that they needed to save and submit all possible conditions, so they simply submitted one condition and left the test. Both cases are consistent with the response data of frequent wrong answers submitted (see Figure 4), but they point to different causes of failing the test.

Since the numbers of conditions saved are different across score groups, comparing the execution time, which is the sum of the duration of different numbers of actions, is not enough to reflect whether the efficiency of high- or low-performing students is similar. We need to further compare the mean execution time per answering event. The full score students spent less time (see Table 6) on conducting each answering related action than the low-performing students. This implies that although the full score students conducted more actions, they were more efficient, by putting less effort on each action, than the lowest score students (and those having partial scores). In this sense, the results in the two tests are consistent: the students who properly apply the control-of-variable strategies show more goal-directed and efficient behaviors [30] than those who failed to do so.

The contrasting results of execution time between the fair and exhaustive tests reveal the differences between the two types of scientific inquiry practice. Although both tests require controlling variables under investigation, the nature of control is different, so are the required cognitive resources to properly complete the tests. In the fair test, to study the effect of a target variable (tree type, see Figure 1), students need to keep the other variable (tree position) unchanged. In the exhaustive test, students need to combine different values (use or not use, see Figure 2) of the variables (fertilizers A and B) to set up a set of conditions for comparison. Properly completing this test requires mentally constructing the conditions not easily foreseen and spending time and energy in thinking and setting up each possible condition, thus requiring more cognitive resources than the fair test, the latter of which only requires adjusting the target variable and holding the other one(s) constant. These results indicate that the same control-of-variables strategy manifests differently in different scientific inquiry practices. Systematic teaching and learning of this strategy require task-specific training in different situations.

All the results reveal the aspects in which high-performing students excel low-performing ones, including: (a) grasping instructions, (b) extracting requirements, and (c) constructing answers. Compared to high-performing students, low-performing students had lower efficiency in grasping necessary knowledge and applying required strategies in the tests. As a consequence, in the fair test, low-performing students struggled in selecting and revising answers, and ended up submitting wrong answers; and in the exhaustive test, they failed to envision all possible conditions, and failed to construct enough conditions as the final answers.

The above discussions focus primarily on statistical differences between the full and lowest score students. This is because that our statistical analyses report consistent results between the two

score groups. However, results are not consistent when partial score groups are involved. Such inconsistency could be due to several reasons. First, some partial score groups contained fewer students than the other two groups. Second, according to the scoring rubrics, the response difference between the full (or the lowest) score and a partial score is smaller than that between the full and lowest scores, which may cause smaller difference in answering events and/or their durations. Both factors reduced the statistical power of the analyses. Third, lacking empirical basis, the predefined score rubrics might not clearly differentiate students having different levels of problem solving competency. This issue is beyond the scope of the current study. Nonetheless, such inconsistency calls for statistically more powerful process-based features to reveal the differences between students having good and poor performances in science inquiry practice and understand how they apply required skills in such practice.

5. CONCLUSIONS

This study makes use of three process-based, temporal measures to analyze how students conduct scientific inquiry in practice. We identify both the global (e.g., durations of thinking, and total duration of execution) differences and local (e.g., execution efficiency) consistency between students who can appropriately apply the control-of-variables strategies in scientific inquiry practice and those who fail to do so. The findings provide new evidence to the general discussions of the relations among individual capacity (e.g., control-of-variables strategy), nature of test (e.g., fair or exhaustive test), problem-solving process (e.g., duration and efficiency of activities), and assessment performance (e.g., submitted answers and scores). The process-based features have proven values in revealing performance differences in the fair and exhaustive tests. Analysis results based on these measures reveal the aspects or stages during the problem-solving process in which teachers can provide guidance or students can self-improve to teach the required inquiry skills or properly apply them, thus improving students' performances in science inquiry practice.

6. ACKNOWLEDGMENTS

This work was supported in part by National Center for Education Statistics (NCES) and Educational Testing Service (ETS). We thank Kathleen Scalise, Madeleine Keehner, Gary Feng, and Christopher Agard from ETS for support on this work. We also thank the anonymous reviewers of EDM 2020 for their valuable comments on this work.

7. REFERENCES

- [1] National Research Council. 1996. *National Science Education Standards*. The National Academies Press, Washington, DC. DOI=<https://doi.org/10.17226/4962>.
- [2] National Assessment Governing Board. 2015. *Science Framework for the 2015 National Assessment of Educational Progress*. Washington, DC. <https://www.nagb.gov/naep-frameworks/science/2015-science-framework.html>.
- [3] National Research Council. 2012. *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. Washington, DC. DOI=[10.17226/13165](https://doi.org/10.17226/13165).
- [4] Bybee, R. W. 2000. Teaching science as inquiry. In *Inquiring into Inquiry Learning and Teaching in Science*, J. Minstrell and E. H. van Zee, Eds. American Association for the Advancement of Science, Washington, D.C., 21–46.

- [5] National Research Council. 2013. *Next Generation Science Standards: For States, by States*. The National Academies Press, Washington, D.C. <https://www.nap.edu>.
- [6] Rönnebeck, S., Bernholt, S., and Ropohl, M. 2016. Searching for a common ground – A literature review of empirical research on scientific inquiry activities. *Studies of Science Education*, 52(2), 161–197. DOI=[10.1080/03057267.2016.1206351](https://doi.org/10.1080/03057267.2016.1206351).
- [7] Mislevy, R. J., Chudowsky, N., Draney, K., Fried, R., Gaffney, T., Haertel, G., et al. 2003. *Design Patterns for Assessing Science Inquiry* (PADI Technical Report 1). Menlo Park, CA. <https://padi.sri.com>.
- [8] Scalise, K. 2014. *Assessment System Design Options for the Next Generation Science Standards (NGSS): Reflections on Some Possible Design Approaches*. ETS, Princeton, NJ. https://www.ets.org/research/policy_research_reports/publications/paper/2014/jvha.
- [9] Chen, Z. and Klahr, D. 1999. All other things being equal: acquisition and transfer of the control-of-variables strategy. *Child Development*, 70(5), 1098–1120. DOI=[10.1111/1467-8624.00081](https://doi.org/10.1111/1467-8624.00081).
- [10] Tschirgi, J. E. 1980. Sensible reasoning: A hypothesis about hypotheses. *Child Development*, 51(1), 1–10. DOI=[10.2307/1129583](https://doi.org/10.2307/1129583).
- [11] Kuhn, D. and Dean, D. 2005. Is developing scientific thinking all about learning to control variables? *Psychological Science*, 16, 866–870. DOI=[10.1111/j.1467-9280.2005.01601628.x](https://doi.org/10.1111/j.1467-9280.2005.01601628.x).
- [12] Montgomery, D. C. 2000. *Design and Analysis of Experiments*, 5th ed. Wiley Text Books, Indianapolis, IN.
- [13] Black, R. 2007. *Pragmatic Software Testing: Becoming an Effective and Efficient Test Professional*. Wiley, New York.
- [14] Klahr, D. and Nigam, M. 2004. The equivalence of learning paths in early science instruction: Effects of direct instruction and discovery learning. *Psychological Science*, 15, 661–667. DOI=[10.1111/j.0956-7976.2004.00737.x](https://doi.org/10.1111/j.0956-7976.2004.00737.x).
- [15] Harrison, A. M. and Schunn, C. D. 2004. The transfer of logically general scientific reasoning skills. In *Proceedings of the 26th Annual Meeting of the Cognitive Science Society*, K. Forbus, D. Gentner, and T. Regier, Eds. Lawrence Erlbaum Associates, Mahwah, NJ, 541–546.
- [16] Kim, M. C., Hannafin, M. J., and Bryan, L. A. 2007. Technology-enhanced inquiry tools in science education: An emerging pedagogical framework for classroom practice. *Science Education*, 91(6), 1010–1030. DOI=[10.1002/sc.20219](https://doi.org/10.1002/sc.20219).
- [17] Shimoda, T. A., White, B. Y., and Frederiksen, J. R. 2002. Student goal orientation in learning inquiry skills with modifiable software advisors. *Science Education*, 86(2), 244–263. DOI=[10.1002/sc.10003](https://doi.org/10.1002/sc.10003).
- [18] Songer, N. B., Lee, H. S., and Kam, R. 2002. Technology-rich inquiry science in urban classrooms: What are the barriers to inquiry pedagogy? *Journal of Research in Science Teaching*, 39(2), 128–150. DOI=[10.1002/tea.10013](https://doi.org/10.1002/tea.10013).
- [19] Ebenezer, J., Kaya, O. N., and Ebenezer, D. L. 2011. Engaging students in environmental research projects: Perceptions of fluency with innovative technologies and levels of scientific inquiry abilities. *Journal of Research in Science Teaching*, 48(1), 94–116. DOI=[10.1002/tea.20387](https://doi.org/10.1002/tea.20387).
- [20] Gobert, J. D., Sao Pedro, M. A., Baker, R. S. J. D., Toto, E., and Montalvo, O. 2012. Leveraging educational data mining for real-time performance assessment of scientific inquiry skills within microworlds. *Journal of Educational Data Mining*, 4(1), 104–143. DOI=[10.5281/zenodo.3554645](https://doi.org/10.5281/zenodo.3554645).
- [21] Taasobshirazi, G., Zuiker, S. J., Anderson, K. T., and Hickey, D. T. 2006. Enhancing inquiry, understanding, and achievement in an astronomy multimedia learning environment. *Journal of Science Education and Technology*, 15(5), 383–395. DOI=[10.1007/s10956-006-9028-0](https://doi.org/10.1007/s10956-006-9028-0).
- [22] Geier, R., Blumenfeld, P. C., Marx, R. W., Krajcik, J. S., Fishman, B., Soloway, E., and Clay-Chambers, J. 2008. Standardized test outcomes for students engaged in inquiry-based science curricula in the context of urban reform. *Journal of Research in Science Teaching*, 45(8), 922–939. DOI=[10.1002/tea.20248](https://doi.org/10.1002/tea.20248).
- [23] Minner, D. D., Levy, A. J., and Century, J. 2010. Inquiry-based science instruction—what is it and does it matter? Results from a research synthesis years 1984 to 2002. *Journal of Research in Science Teaching*, 47(4), 474–496. DOI=[10.1002/tea.20347](https://doi.org/10.1002/tea.20347).
- [24] Pedaste, M., Mäeots, M., Siiman, L. A., de Jong, T., van Riesen, S. A. N., Kamp, E. T., Manoli, C. C., Zacharia, Z. C., and Tsourlidaki, E. 2015. Phases of inquiry-based learning: Definitions and the inquiry cycle. *Education Research Review*, 14, 47–61. DOI=[10.1016/j.edurev.2015.02.003](https://doi.org/10.1016/j.edurev.2015.02.003).
- [25] Wilson, C. D., Taylor, J. A., Kowalski, S. M., and Carlson, J. 2010. The relative effects and equity of inquiry-based and commonplace science teaching on students’ knowledge, reasoning, and argumentation. *Journal of Research in Science Teaching*, 47(3), 276–301. DOI=[10.1002/tea.20329](https://doi.org/10.1002/tea.20329).
- [26] Dostál, J. 2015. Theory of problem solving. *Procedia-Social and Behavioral Sciences*, 174(1), 2798–2805. DOI=[10.1016/j.sbspro.2015.01.970](https://doi.org/10.1016/j.sbspro.2015.01.970).
- [27] Dixon, W. J. 1960. Simplified estimation from censored normal samples. *The Annals of Mathematical Statistics*, 31(2), 385–391. DOI=[10.1214/aoms/1177705900](https://doi.org/10.1214/aoms/1177705900).
- [28] Kruskal, W. H. and Wallis, W. A. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260), 583–621. DOI=[10.2307/2280779](https://doi.org/10.2307/2280779).
- [29] R Core Team. 2019. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. <http://r-project.org>.
- [29] Arslan, B., Keehner, M., Jiang, Y., Gong, T., Katz, I. R., and Yan, F. 2020. The effect of drag-and-drop item features on test-taker performance and response strategies. *Educational Measurement: Issues and Practices*. DOI=[10.1111/emip.12326](https://doi.org/10.1111/emip.12326).
- [30] Shimoda, T. A., White, B. Y., and Frederiksen, J. R. 2002. Student goal orientation in learning inquiry skills with modifiable software advisors. *Science Education*, 86(2): 244–263. DOI=[10.1002/sc.10003](https://doi.org/10.1002/sc.10003).

Confident Learning Curves in Additive Factors Modeling

Cyril Goutte
National Research Council Canada
Cyril.Goutte@nrc-cnrc.gc.com

Guillaume Durand
National Research Council Canada
Guillaume.Durand@nrc-cnrc.gc.ca

ABSTRACT

Learning curves are an important tool in cognitive diagnostics modeling to help assess how well students acquire new skills, and to refine and improve knowledge component models. Learning curves are typically obtained from a model estimated on real data obtained from a finite, and usually limited, sample of students. As a consequence, there is some uncertainty associated with estimating the model from that sample, and a risk that the inferences made using learning curves derived from the estimated model are over-confident one way or another. Based on previous work modeling the uncertainty on Additive Factors Model parameters, we derive a principled way to quantify the confidence in learning curves associated with each knowledge component. We show that our approach leads to relatively tight bounds on the learning curves, much tighter than a naive approach relying only on parameter uncertainty. This also reveals a disparity across knowledge components regarding how confident one can be in how well these skills are mastered.

Keywords

Learning Curves, Additive Factors Modeling, Knowledge Cognitive Diagnostics Model

1. INTRODUCTION

Learning curves are a crucial tool for cognitive diagnostics modeling. They help build relevant competency frameworks to accurately measure learners skills and to give them meaningful guidance and feedback in intelligent tutoring systems (ITSs). More precisely, learning curves measure the rate at which students, or simulated artefacts [22], acquire competencies. This allows to evaluate the suitability of a competency framework (aka *Q-matrix*) and a principled comparison of different learning systems. Learning curves are “graphs that plots performance on a task versus the number of opportunities to practice” [17]. In the educational field, learning curves usually take as learning performance metric the error rate (or equivalently success rate) when applying

an individual skill or a set of skills. They were empirically found to follow a “power law of practice” [18], which means that the error rate over time decreases roughly linearly with the logarithm of the number of practice trials taken (aka *opportunities*). Comparing ITSs or sections of ITS can be done by considering the steepness of the curve: A steeper curve indicates a faster acquisition of the skills practiced [17].

However, tracking the performance of skills learned in a multidimensional learning environment can be difficult, as those environments combine different set of skills evaluated together. In such situations, some cognitive diagnostic models can be useful to compare learning systems but also to understand the learning mechanisms at play [10]. The Additive Factors Model (AFM) [1], a well known cognitive diagnostics model, does this by assuming that each necessary skill in an item comes with a skill-specific additive contribution towards the probability of success on the item. Fitted AFM parameters can also be used to draw learning curves that compensate for the *attrition bias* [9]: Over time, fewer learners tend to practice some items because many of them have learned the skill, and the curves tend to quickly degenerate, impacting the estimates of the slopes and the diagnostics of how much learning has occurred. In addition, when learning curves are drawn directly from AFM parameters, the validity of the inferences that can be made will depend greatly on the reliability of the parameters values, and ultimately on the quality of the fitted data. More precisely, fitted parameter values tend to compensate for noise, missing values (e.g. due to *attrition*) or mis-specified competency models. Rupp and Templin [21] showed for instance how the fitted values of model parameters in DINA [11] would inflate when fitted with purposely erroneous Q-matrices. We can expect a similar impact with any model using Q-matrices, including AFM, a situation made worse by the fact that, in reality, perfect Q-matrices are difficult to identify [5], even when they are retro-engineered from performance data [19]. This motivates the necessity to estimate not only parameter values, but also the statistical confidence on those values, and take into account this uncertainty in any model interpretation, whether based on those values or on the associated learning curves.

Previous work investigated the estimation of standard errors on DINA [20] or AFM [7] parameters, and showed how it could impact learning curves shape and ultimately AFM interpretability and usefulness [15]. Assuming independence across parameters, they produced bounds on learning curves

Cyril Goutte and Guillaume Durand "Confident Learning Curves in Additive Factor Modeling" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 424 - 430

using standard confidence intervals on parameter values. However, in practice, the AFM skills parameters (Section 2) are clearly not independent. In this contribution, we show how we can take into account the structure of the covariance between the AFM parameters in order to better model and control the uncertainty on those parameters. We describe a technique for generating confidence intervals on the learning curves using a sampling approach. We illustrate how this works on several competency models from a well-known dataset obtained from a geometry tutoring course, and we show how it allows us to compare how different competency models may model the same skills with different confidence.

In the following Section, we quickly describe the AFM model and introduce our method for obtaining more adequate estimates of the confidence intervals on the learning curves. Section 3 quickly describes the well known EDM dataset that we experiment with in Section 4. Section 5 discusses the results and their impact before we conclude.

2. METHOD

The Additive Factors Model (AFM) introduced by Cen et al. [1, 3] is used in the PSLC-Datashop [12] in order to evaluate domain models. It models the probability of success of a student i on item j using user and skill specific parameters:

$$P(Y_{ij} = 1 | \alpha_i, \beta, \gamma) = \sigma \left(\alpha_i + \sum_{k=1}^K \beta_k q_{jk} + \sum_{k=1}^K \gamma_k q_{jk} t_{ik} \right) \quad (1)$$

with $\sigma(x) = 1/(1 + e^{-x})$ the logistic function, and

α_i is the *proficiency* of student i ,
 β_k is the *easiness* of skill $k = 1 \dots K$,
 γ_k is the *learning rate* for skill k ,
 $\mathbf{Q} = [q_{jk}]$ is the $J \times K$ *Q-matrix*, representing the cognitive model mapping items to skills,
 t_{ik} is the number of times student i has practiced skill k (on any item).

Parameters $\theta = (\alpha, \beta, \gamma)$ are estimated by maximizing the (penalized) likelihood of the model over observed student outcomes (see e.g. [6]). One attractive feature of AFM is that it easily provides performance curves showing how students acquire skills. Among the different types of learning curves that can be derived from AFM [9, 8], we focus on the data- and student-independent *idealized learning curve* [8],¹ that simply traces the probability of error for an idealized student with $\alpha = 0$ proficiency, on an item with a single skill k :

$$LC_k(t) = 1 - P(Y = 1 | \alpha = 0, \beta, \gamma) = \sigma(\beta_k + \gamma_k t). \quad (2)$$

Learning curves are typically computed with the maximum penalized likelihood parameters $\hat{\theta} = (\hat{\alpha}, \hat{\beta}, \hat{\gamma})$. As noted for example by Philipp et al. [20] and derived for AFM by Durand et al. [7], one can also estimate the uncertainty on $(\hat{\alpha}, \hat{\beta}, \hat{\gamma})$, in the form of standard errors. This is relatively straightforward as the covariance matrix on parameter estimates is asymptotically equal to the inverse of the information matrix, $\text{Cov}(\hat{\theta}) = \mathcal{I}_{\hat{\theta}}^{-1}$. The information matrix $\mathcal{I}_{\hat{\theta}}$ can

¹aka Individual Learning Curve in [9].

Algorithm 1: Error bars on learning curve for skill k .

Data: Parameters $\hat{\theta}$, covariance $\text{Cov}(\hat{\theta})$

Parameters: Target skill k , simulation sample size N

Result: Error bars for the learning curve for skill k , at a set of opportunities $\{t = 1 \dots T\}$

repeat

 Sample $\theta^{(i)} \sim \mathcal{N}(\hat{\theta}, \text{Cov}(\hat{\theta}))$;

 Compute learning curve $LC_k^{(i)}(t)$ for target skill k

until N simulations;

For each opportunity t , compute confidence interval $[\ell_k(t), u_k(t)]$ using relevant quantiles² of $\{LC_k^{(i)}(t)\}$.

be estimated from first or second order derivatives of the cost function [20, eq. 3, 4]. This also provides a key to quantifying the uncertainty on the learning curves. Using the fact that parameters are (asymptotically) normally distributed around $\hat{\theta}$ with the known covariance matrix $\text{Cov}(\hat{\theta})$ [7], we can sample sets of parameters from that multivariate Gaussian distribution, compute the learning curve for each set of parameters, then empirically estimate the error bars on the learning curve through the relevant quantile statistics, as outlined in Algorithm 1.

Although Algorithm 1 focuses on producing error bars on the learning curves, we can also use the simulated sample to evaluate the stability of the entire learning curve, using for example the average standard deviation across opportunities:

$$\bar{\sigma}_k = \frac{1}{T} \sum_{t=1}^T \text{st.dev.}\{LC_k^{(i)}(t)\}$$

Lower $\bar{\sigma}_k$ indicate that the sampled learning curves are closer together, thus the learning curve is more stable.

3. DATA

For our experiments, we used the “Geometry Area (1996-97)”, a public dataset from DataShop [12]. This dataset contains 6778 observations of the performance obtained by 59 students completing 139 unique items from the “area unit” of the Geometry Cognitive Tutor course (school year 1996-1997). This dataset has been extensively used [1, 2, 7, 13, 14]. We selected three knowledge component (KC) models:

- hLFASearchAICWholeModel3arith0 (referred to simply as **arith** below),
- hLFASearchModel1-context (**context** below),
- Original (**orig** below).

These KC models were selected for their reasonable numbers of skills and observations but also because they have distinctive goodness of fit metrics, suggesting that they are high-performing KC models. Table 1 shows that the best predictive model would be **arith**. The number of skills (KCs) seems to have limited impact on the goodness of fit metrics.

²For example, the 95% confidence interval is obtained as

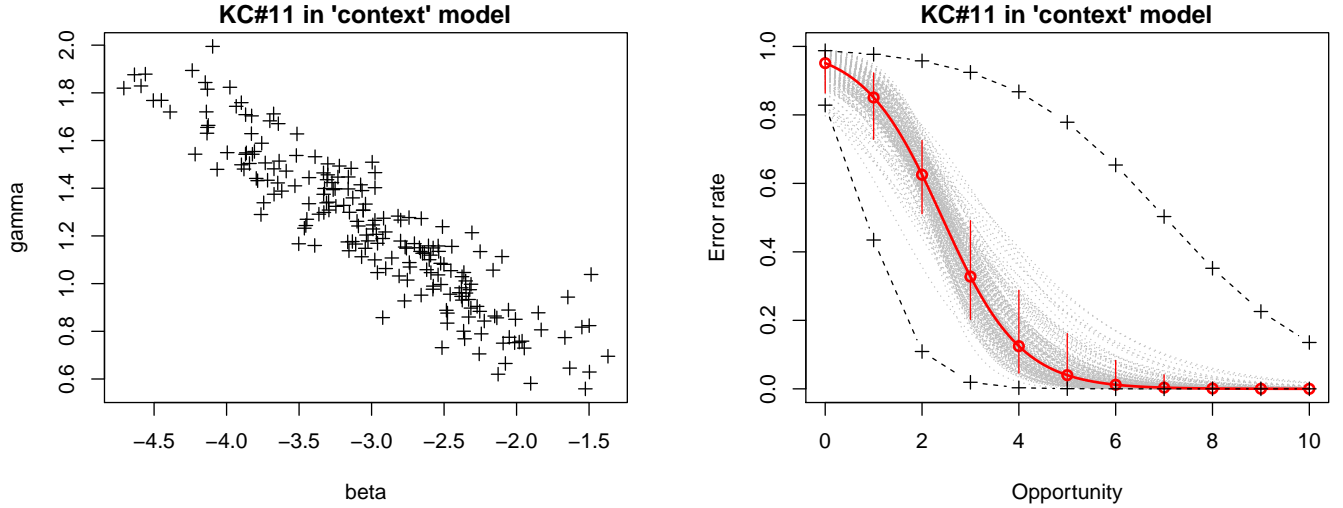


Figure 1: Left: Sampled β and γ for KC#11 of the context model. Right: Corresponding learning curves (in light gray); the LC given by the AFM model is in red, with 95% confidence intervals at opportunities up to 10 shown as red vertical bars. The 95% CI from [7] is indicated in black crosses for comparison.

Table 1: Characteristics and predictive quality of the KC models, as computed by PSLC-Datashop.

Name	KCs	Stud.	#Obs.	AIC	BIC	RMSE
arith	18	59	5104	4948	5569	.397
context	12	59	5104	5030	5573	.399
orig	15	59	5104	5180	5762	.407

Another motivation for choosing these KC models is their skills sharing as some skills have an identical mapping to items in another model, allowing to compare the stability of the same skill across KC models.

4. EXPERIMENTS

In this section, we first illustrate how we derive error bars on the learning curve for a specific KC, then show results for an entire KC model, and finally we compare the stability of learning curves for equivalent skills in different KC models.

4.1 Illustration

We focus on KC#11 (*equi-tri-height-from-base/side*) from KC-model *context*. This is a relatively hard ($\beta = -2.97$) skill, but with quick learning ($\gamma = 1.23$). Figure 1 (left) shows the values of β_{11} and γ_{11} that were sampled by Algorithm 1 for this KC. As seen in the plot, the marginal uncertainty on β_{11} and γ_{11} is quite high (from -4.5 to -1.5 for β_{11}), but they are also very correlated: samples with higher *easiness* have lower *learning rate*.

Each of the points in Fig. 1 (left) is translated into a corresponding learning curve (Eq. 2) in dotted light gray in Fig. 1 (right). Due to the correlation noted before, we can see that the sampled learning curves are actually fairly stable, compared to what extremes of the distributions of β_{11} and γ_{11} would suggest (see dashed lines with crosses in Fig. 1,

$[q_{2.5}, q_{97.5}]$, where q_ϵ is such that $\epsilon\%$ of the sample is below q_ϵ and $(100 - \epsilon)\%$ is above.

which replicates Fig. 4 from [7]). The red curve in Figure 1 (right) is the learning curve computed from the AFM solution, with 95% confidence intervals obtained from the sample at each opportunity indicated as red bars. We see that although there is some uncertainty around the steep part of the curve, the learning curve is well-controlled and easy to diagnose, indicating that the skill is completely acquired after around 5 opportunities.

4.2 Application to KC models

We now show how we can generate learning curves with confidence intervals for a full KC model. The process illustrated above is applied to each KC, producing one learning curve with confidence bounds. For improved readability, we show the results on KC-model *context*, which has the smallest number of KCs among our three models.

Figure 2 shows the learning curves for the twelve knowledge components. We can see that most learning curves are well-controlled. The average standard deviation $\bar{\sigma}$, depending on the skill, ranges from 2% to 8%. "Flat" KCs tend to have lower uncertainty, which is understandable: when the error rate for a skill is low and flat, this is easy for the model to pick up with confidence by predicting high success (high β) for that skill.

4.3 Comparison of KC models

By better estimating and controlling the uncertainty in learning curves, we can more reliably compare how skills are acquired according to different KC models.

In Figure 3 we show the same skill, *compose-by-multiplication*, as modeled by the 12-skill model *context*, and by the 15-skill model *orig*. The shapes of the learning curves are very similar, which is not surprising as both KCs are associated to the same items, and estimated from the same student outcomes. Despite differences due to the influence of other KCs in the models, the resulting values of β and γ are similar.

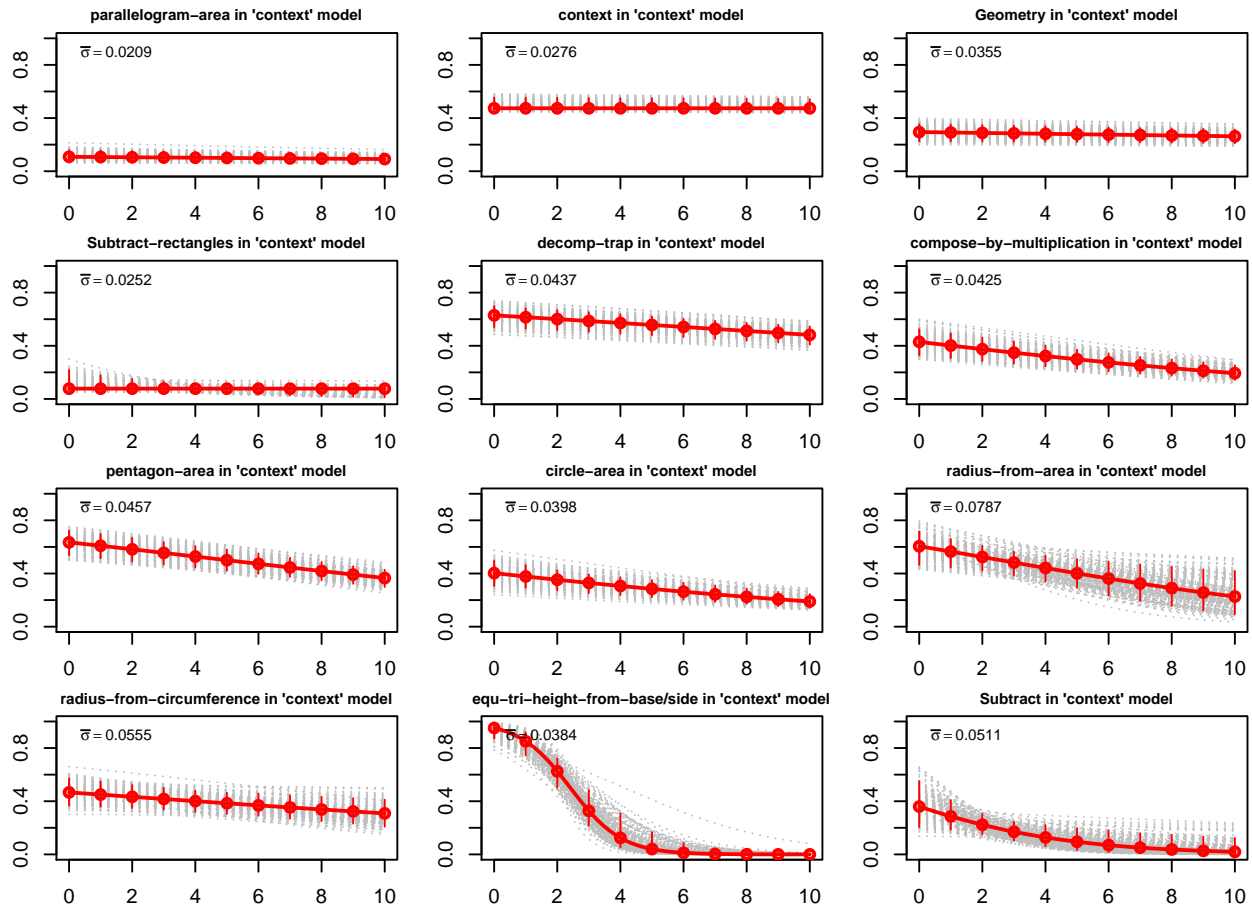


Figure 2: All learning curves with confidence intervals for KC model context.

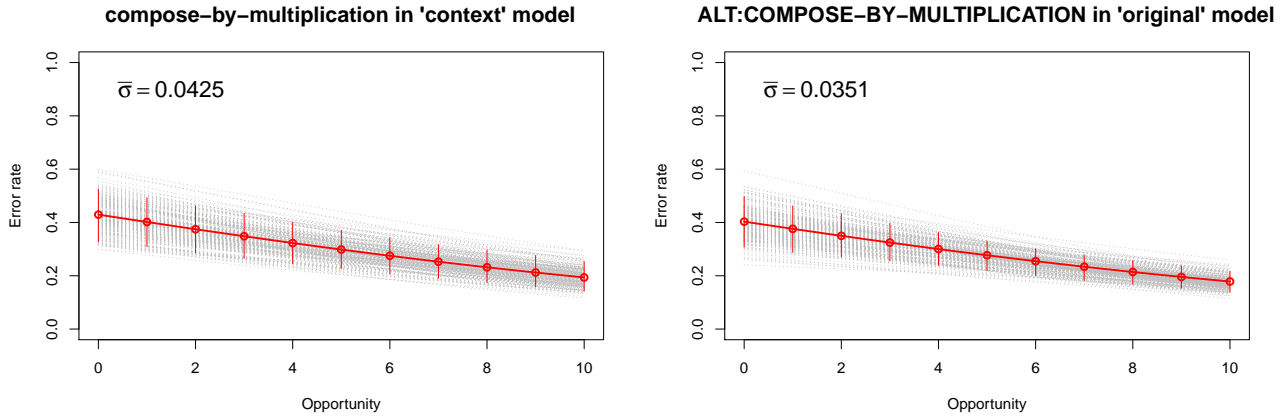


Figure 3: KC compose-by-multiplication from KC models context (left) and orig (right). $\bar{\sigma}$ is the average uncertainty across opportunities (lower is better).

The error bars, however, show that the confidence is slightly better in the *orig* model, showing an average dispersion of around 3.5% error across the learning curve (versus 4.3% in *context*). This shows that even in a model with more KCs, learning curves can be modelled with higher confidence.

Our second example, in Figure 4, compares similar skills, *compose-subtract* from *arith*, and *Subtract* from *orig*. Again, the general shape of the learning curves are similar, due to similar values for the estimated β and γ in each model.³ The sampled learning curves also seem quite similar, sug-

³For *arith*, $\beta = .588 \pm .524$ and $\gamma = .329 \pm .200$, while for

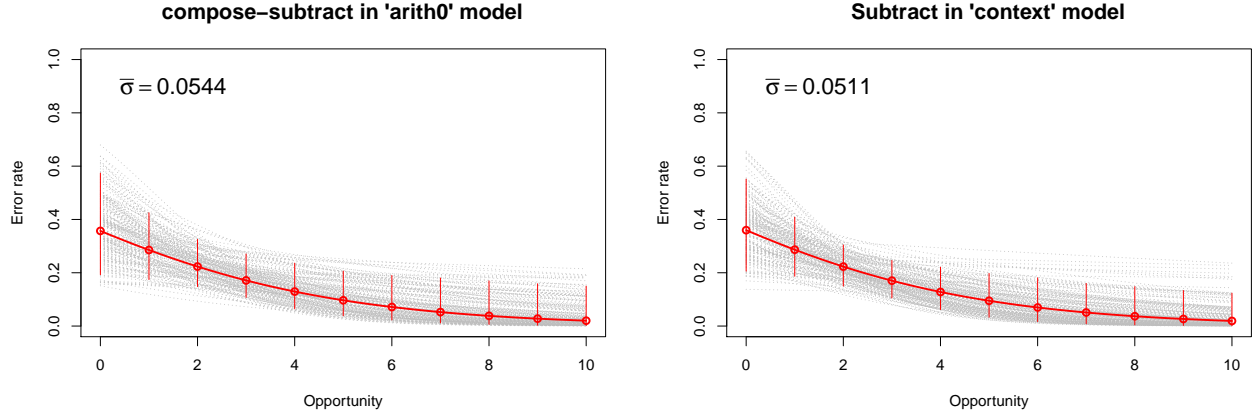


Figure 4: KC compose-subtract from model arith (left) and KC Subtract from orig (right). $\bar{\sigma}$ is the average uncertainty across opportunities (lower is better).

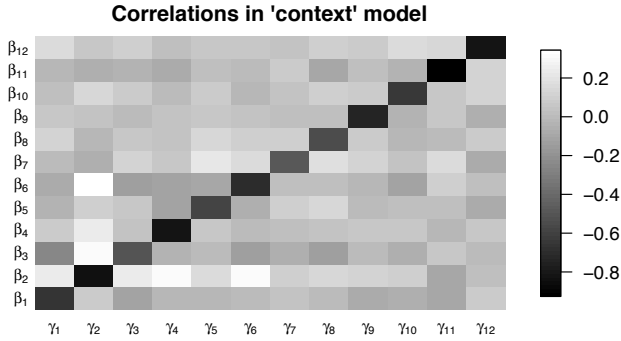


Figure 5: Structure of the correlation between β (y-axis) and γ (x-axis) for all KCs in model context.

esting that both KC models represent that skill with similar levels of confidence. This is confirmed by the value of the average dispersion, which is 5.4% for one model and 5.1% for the other. We see again that the different number of KCs has limited impact on how confident the models are on a particular skill.

5. DISCUSSION

Figure 1 (left) showed that there is a strong correlation between the sampled values of β_{11} and γ_{11} . The impact of this correlation on the actual learning curve is that, according to the model, this knowledge component can be modeled by a higher easiness (starting with lower error) and lower learning rate (flatter curve), or by a lower easiness and higher learning rate (i.e. starting higher but dropping faster). This finding actually generalizes to the entire KC model, as shown by the correlation matrix in Figure 5. We see that there is a consistently strong *negative correlation* between the β and γ parameters for each knowledge component, due to this compensatory mechanism. There are also some correlations between parameters of different KC, which may suggest some compensatory effects in the AFM model.

context, $\beta = .576 \pm .523$ and $\gamma = .336 \pm .200$.

One straightforward outcome of this work is that the proposed method provides a much better estimate of the confidence in a learning curve than the method proposed in [7], which relied on the marginal distribution of AFM parameters β and γ and used the boundaries of straight confidence intervals on each parameter independently. We included their 95% confidence interval as black crosses in Figure 1: that suggests that the uncertainty on the learning curve is high up to 8 or more opportunities. By contrast, our approach shows that the actual uncertainty is much better controlled, and that the skill is essentially learned by opportunity 5 or 6.

In this paper, we have worked with the basic learning curve called the *individual learning curve* in [9] or the *idealized learning curve* in [8]. We note that this work can be applied to any learning curve that relies on the parameters of the AFM model. This includes in particular the *completed learning curve* [9], where empirical observations of success/failure are completed by model estimates.

In previous work, Harpstead and Aleven [10] used empirical learning curve analysis to inform educational game design. They derive empirical curves and AFM-fitted curves, with standard errors on the curves, using a completely different approach from ours. Contrary to the approach advocated here, which relies on the core uncertainty on model parameters resulting from a maximum (penalized) likelihood estimation, their learning curves and error bars are obtained using non-parametric smoothing (LOESS [4], presumably from the `stat-smooth` function of the `ggplot2` R package). On the empirical measurements of success, this produces learning curves that are based on observations alone, and therefore may not have the desirable properties enforced by the AFM model, such as monotonicity (decreasing learning curves). On the fitted AFM predictions, those properties are enforced and apparent from the learning curves.⁴ Two key differences with our approach, however, are:

1. The use of fitted AFM values to produce error rate

⁴Blue curves in [10], Figs 3, 4 and 7.

predictions does not take into account the uncertainty in parameter values due to estimation from a finite sample, and

2. The width of the error bars are directly impacted by the number of students at each opportunity, typically resulting in widening error bars as attrition kicks in. By contrast our sampling-based algorithm often yields narrowing error bars as opportunities increase and the error rates near zero (for all sampled parameters).

A more systematic study of differences between our approach and the non-parametric smoothing of model estimates would require further study. The opportunity of combining both approaches in order to take into account the uncertainty due to parameter estimation and sampling uncertainty across the finite set of students seems particularly promising.

6. CONCLUSION

In this contribution, we provided a principled way to estimate and control the confidence in learning curves derived from the Additive Factors Model. Error bars on the learning curves account for the statistical uncertainty associated with estimating the AFM model from a finite set of students. They allow to more accurately and more confidently interpret how skills are acquired by students. We showed how this allows to characterize learning for all skills of a KC model of a geometry tutoring course. We also showed how modeling the confidence of learning curves can help compare how two different KC models represent the same skill. Our approach was illustrated here on one type of learning curve, but it can be applied to any alternative learning curve, as long as it can be computed from the usual AFM parameters. In addition, the same idea can be applied in a straightforward way to any cognitive diagnostic model for which a covariance on parameters can be computed. This includes in particular, models estimated by penalized maximum likelihood. For instance, the Individualized-slope Additive Factors Model (iAFM) [16], that extends AFM with a student learning rate, could be an excellent candidate to our method, especially as authors noticed that iAFM "[student] learning rate is significantly related to estimates of student ability". Finally, our hope is that this work will help spread the use of learning curves with well-controlled confidence among practitioners of AFM.

7. REFERENCES

- [1] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis – A general method for cognitive model evaluation and improvement. In M. Ikeda, K. D. Ashley, and T.-W. Chan, editors, *Intelligent Tutoring Systems: 8th Intl. Conference (ITS 2006)*, pages 164–175, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [2] H. Cen, K. Koedinger, and B. Junker. Is overpractice necessary? Improving learning efficiency with the cognitive tutor through educational data mining. In R. Luckin, K. R. Koedinger, and J. Greer, editors, *Proc. 2007 Conf. on Artificial intelligence in Education: Building Technology Rich Learning Contexts that Work*, number 158 in Frontiers in Artificial Intelligence and Applications, pages 511–518, Amsterdam, Netherlands, 2007. IOS Press.
- [3] H. Cen, K. Koedinger, and B. Junker. Comparing two IRT models for conjunctive skills. In B. P. Woolf, E. Aïmeur, R. Nkambou, and S. Lajoie, editors, *Proc. 9th International Conf. on Intelligent Tutoring Systems (ITS 2008)*, Lecture Notes In Computer Science, pages 796–798, Berlin, Heidelberg, 2008. Springer-Verlag.
- [4] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836, 1979.
- [5] G. Durand, N. Belacel, and C. Goutte. Evaluation of expert-based Q-matrices predictive quality in matrix factorization models. In *Design for Teaching and Learning in a Networked World, EC-TEL 2015 conference*, pages 56–69. Springer, 2015.
- [6] G. Durand, C. Goutte, N. Belacel, Y. Bouslimani, and S. Léger. Review, computation and application of the additive factor model (AFM). Tech. Report 23002483, National Research Council Canada, 2017.
- [7] G. Durand, C. Goutte, and S. Léger. Standard error considerations on AFM parameters. In K. E. Boyer and M. Yudelson, editors, *Proc. 11th International Conf. on Educational Data Mining (EDM 2018)*. International Educational Data Mining Society (IEDMS), 2018.
- [8] T. Effenberg, R. Pelánek, and J. Čechák. Exploration of the robustness and generalizability of the additive factors model. In *Proceedings of LAK'20*, 2020.
- [9] C. Goutte, G. Durand, and S. Léger. On the learning curve attrition bias in additive factor modeling. In C. Penstein Rosé, R. Martínez-Maldonado, H. U. Hoppe, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, B. McLaren, and B. du Boulay, editors, *Artificial Intelligence in Education*, pages 109–113. Springer, 2018.
- [10] E. Harpstead and V. Aleven. Using empirical learning curve analysis to inform design in an educational game. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, page 197–207, New York, NY, USA, 2015. Association for Computing Machinery.
- [11] B. W. Junker and K. Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(3):258–272, 2001.
- [12] K. Koedinger, R. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the EDM community: The PSLC Datashop. In C. Romero, S. Ventura, M. Pechenizkiy, and R. Baker, editors, *Handbook of Educational Data Mining*. CRC Press, 2010.
- [13] K. R. Koedinger, K. Cunningham, A. Skogsholm, and B. Leber. An open repository and analysis tools for fine-grained, longitudinal learner data. In *The 1st International Conf. on Educational Data Mining (EDM 2008)*, pages 157–166, 2008.
- [14] K. R. Koedinger, E. A. McLaughlin, and J. C. Stamper. Automated student model improvement. In *EDM*, pages 17–24. www.educationaldatamining.org, 2012.

- [15] R. Kop, H. Fournier, and G. Durand. A Critical Perspective on Learning Analytics and Educational Data Mining. In C. Lang, G. Siemens, A. F. Wise, and D. Gašević, editors, *The Handbook of Learning Analytics*, pages 319–326. Soc. for Learning Analytics Research (SoLAR), Alberta, Canada, 2017.
- [16] R. Liu and K. R. Koedinger. Towards reliable and valid measurement of individualized student parameters. In X. Hu, T. Barnes, A. Hershkovitz, and L. Paquette, editors, *Proc. 10th International Conf. on Educational Data Mining, (EDM 2017)*. International Educational Data Mining Society (IEDMS), 2017.
- [17] B. Martin, A. Mitrovic, K. R. Koedinger, and S. Mathan. Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-Adapted Interaction*, 21(3):249–283, Aug 2011.
- [18] A. Newell and P. S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. *Cognitive skills and their acquisition*, 1:1–55, 1981.
- [19] H. Nguyen, Y. Wang, J. C. Stamper, and B. M. McLaren. Using knowledge component modeling to increase domain understanding in a digital learning game. In M. C. Desmarais, C. F. Lynch, A. Merceron, and R. Nkambou, editors, *Proc. 12th Intl. Conf. on Educational Data Mining, (EDM 2019)*. International Educational Data Mining Society (IEDMS), 2019.
- [20] M. Philipp, C. Strobl, J. de la Torre, and A. Zeileis. On the estimation of standard errors in cognitive diagnosis models. *Journal of Educational and Behavioral Statistics*, 2017.
- [21] A. A. Rupp and J. Templin. The effects of Q-matrix misspecification on parameter estimates and classification accuracy in the dina model. *Educational and Psychological Measurement*, 68(1):78–96, 2008.
- [22] D. Weitekamp III, E. Harpstead, C. J. MacLellan, N. Rachatasumrit, and K. R. Koedinger. Toward near zero-parameter prediction using a computational model of student learning. In M. C. Desmarais, C. F. Lynch, A. Merceron, and R. Nkambou, editors, *Proc. 12th Intl. Conf. on Educational Data Mining, (EDM 2019)*. International Educational Data Mining Society (IEDMS), 2019.

Towards Fair Educational Data Mining: A Case Study on Detecting At-risk Students

Qian Hu
Department of Computer Science
George Mason University
Fairfax, Virginia
qhu3@gmu.edu

Huzefa Rangwala
Department of Computer Science
George Mason University
Fairfax, Virginia
rangwala@cs.gmu.edu

ABSTRACT

Over the past decade, machine learning has become an integral part of educational technologies. With more and more applications such as students' performance prediction, course recommendation, dropout prediction and knowledge tracing relying upon machine learning models, there is increasing evidence and concerns about bias and unfairness of these models. Unfair models can lead to inequitable outcomes for some groups of students and negatively impact their learning. We show by real-world examples that educational data has embedded bias that leads to biased student modeling, which urges the development of fairness formalizations and fair algorithms for educational applications. Several formalizations of fairness have been proposed that can be classified into two types: (i) group fairness and (ii) individual fairness. Group fairness guarantees that groups are treated fairly as a whole, which might not be fair to some individuals. Thus individual fairness has been proposed to make sure fairness is achieved on individual level. In this work, we focus on developing an individually fair model for identifying students at-risk of underperforming. We propose a model which is based on the idea that the prediction for a student (identifying at-risk students) should not be influenced by his/her sensitive attributes. The proposed model is shown to effectively remove bias from these predictions and hence, making them useful in aiding all students.

Keywords

Fairness, at-risk students detection, decision making, student modeling

1. INTRODUCTION

Educational data mining (EDM) approaches seek to analyze student-related data with the objective of improving learning outcomes for students. Many machine learning methods have been proposed for student modeling and forecasting. However, in the past few years, concerns have emerged about the fairness of machine learning models. An investigation by

ProPublica has found that a machine learning tool COMPAS used to predict risk of recidivism exhibits alarming bias against African-American defendants. It shows that the false positive rate of African-American defendants is twice as their white counterparts (45% vs. 23%) [1]. Buolamwini et al. [3] observed imbalanced gender and skin type distributions in facial recognition datasets. Their study shows that facial recognition algorithms are more likely to misclassify darker-skinned females with error rates up to 34.7%, while the maximum error rate for light-skinned males is 0.8%. In health care, an algorithm used to guide health decisions found that African-American patients assigned the same level of risk are sicker than white patients [24].

In the domain of EDM, unfairness has also been observed. In academic performance prediction systems, social indicators have been found to predict low-performance of male students more accurately than that of female students [29]. A study by Doroudi et al. [7] showed that although personalized models were more equitable than treating all students the same, they were still not fair when relying on inaccurate models and the inequities could cascade as the amount of content increases.

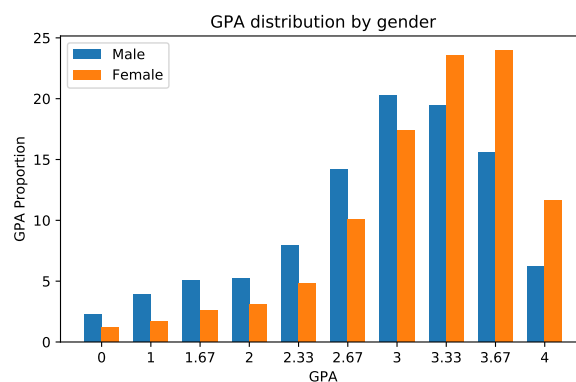


Figure 1: GPA distribution by gender.

Machine learning models learn from data. If bias is recorded in data, models trained on the biased data can also be biased [3]. Bias is also observed in educational data. Figures 1 and 2 show the average GPA of students by gender and race at George Mason University over a period of ten years. The GPA of a student is his/her accumulative GPA as of the last term before graduation. In Figure 1, average GPA of male

Qian Hu and Huzefa Rangwala "Towards Fair Educational Data Mining: A Case Study on Detecting At-risk Students" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 431 - 437

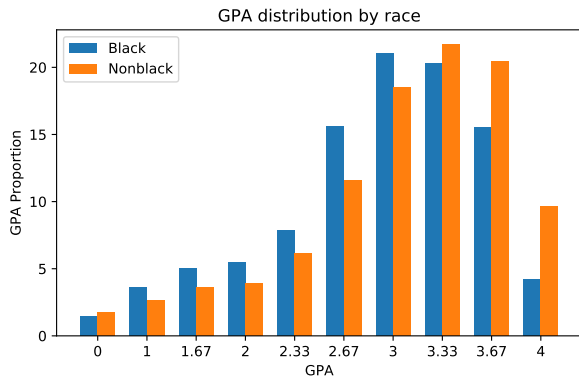


Figure 2: GPA distribution by race.

students is skewed towards lower GPAs, while average GPA of female students is skewed towards higher GPAs. The average GPA of overall female students is 3.15 which is higher than that of male students 2.86. Figure 2 shows the average GPA of African-American and non-African-American students. From the figure, we can observe that average GPA of African-American students leans towards left while that of non-African-American students leans towards right. The data shows that the average GPA of African-American students is 2.86, while it is 3.03 for non-African-American students.

Biased data can lead to biased machine learning models which can be harmful to minority groups. For example, models predicting a group of students to be at-risk or underperforming can discourage them and undermine their learning outcomes. To resolve the harmful results brought about by inequity of machine learning, there are critical needs to develop fair machine learning algorithms.

In this work, we build a fair machine learning model based on metric free individual fairness. Metric free individual fairness assumes that an individual’s qualification should not be changed if his/her sensitive attribute is changed [19]. In this paper, without loss of generality we assume there are two sensitive attributes. The proposed model is composed of two classifiers. Each classifier corresponds to a sensitive group. The classifier corresponding to the individual’s sensitive attribute predicts the individual’s probability of being positive, while the probability of the other classifier indicates the individual’s probability of being positive if his/her sensitive attribute is changed. According to the definition of metric free individual fairness, the two probability distributions should be nearly identical. The proximity of the two probability distributions is treated as fairness. The closer the two distributions, the fairer the prediction is. In addition to fairness, we also care about the accuracy of the classifier. Therefore, the overall objective we seek to optimize is the accuracy of the classifier corresponding to the individual and the proximity of the distributions of the two classifiers.

The proposed model is evaluated on datasets collected from George Mason University and the task is detecting at-risk students. The experimental results show the efficacy of the

proposed model at mitigating bias. Although, the overall data shows that female and non-African-American students have higher overall performance, we observe that the bias is different for different courses. Specifically, in some courses female students belong to disadvantaged group, while in other courses male students are in disadvantaged group. This observation is useful for future work on developing fair machine learning models in educational setting.

The rest of the paper is organized as following. Section 2 discusses related work on EDM and fairness. The following section introduce preliminary on the definition of individual fairness. In Section 4, we propose our fair model for at-risk students detection. Datasets and experimental protocol is described in Section 5. Section 6 presents experimental results and analysis. The last section concludes the paper and discusses future work.

2. RELATED WORK

In this work, we focus on mitigating bias in classification tasks. We first describe related works in EDM that rely on classification. Then we describe the formalizations of fairness. Lastly, we talk about proposed methods for fair machine learning.

2.1 Classification Problems in EDM

In educational data mining, there are many tasks that can be formulated as a classification problem and several prior works have been proposed in this area such as affect detection [30], dropout prediction [4], graduation prediction [20], at-risk student detection [17, 28], knowledge tracing [31], etc.

Affect detection is the task of classifying a student’s affective states such as boredom, confusion, delight, concentration and frustration by using sensor [26] and sensor-free [2] data. Vinayak et al. [15] proposed to predict student dropout using a Naive-Bayes classifier. Ojha et al. [25] proposed SVMs, Gaussian Processes and Deep Boltzmann Machines for student’s graduation prediction using factors such as pre-university preparation. A set of human-interpretable features have been engineered by Polyzou et al. [28] for at-risk student detection. All these tasks can be formulated as a classification problem. However, all these works did not consider the potential bias and discrimination of the models. In this work, we try to build a general method that can be used for different kinds of tasks. To test the proposed method, we focus on the task of identifying at-risk students.

2.2 Fairness Formalizations

Over the years, different formalizations of fairness have been proposed that focus on different aspects. For example, statistical parity [11] requires that the probability of being predicted as positive across all the groups should be nearly the same. Equal odds imposes the constraint that the true positive rate should be the same for all the groups [14]. Equal opportunity requires a qualified individual should be predicted as qualified regardless of his/her sensitive attribute [14]. Another type of fairness formalization focuses more on individual level. The notion of individual fairness proposed by Cynthia et al. [8] assumes that similar individuals should be treated similarly. However, the requirement of a

problem-specific similarity metric limits its adoption [5]. Hu et al. [19] proposed metric free individual fairness based on the assumption that the prediction outcome of an individual should not be influenced by the individual's sensitive attribute. The elimination of similarity metric makes implementation of metric free individual fairness easier.

2.3 Fair Machine Learning Algorithms

Several algorithms have been proposed to achieve individual fairness. Based on John Rawls' notion of fair equality of opportunity, Joseph et al. [21] proposed an individual fairness notion that a worse individual should never be favored over a better one. The unfairness comes from the prediction's dependence on sensitive attribute. To remove the dependence, Zemel et al. [32] proposed learning a fair representation which does not contain sensitive information. The representation is a cluster of embedding vectors. Following the idea of learning fair representation, Edwards [9] proposed to remove sensitive information from the learned representation by using adversarial learning. The input feature vectors are mapped to an embedding vector by an encoder. An adversary tries to predict the sensitive attribute from the representation. The encoder and the adversary plays a minimax game to remove sensitive information. The fair representation learning algorithms achieve individual fairness by first learning a representation and then training a classifier based on the learned representation. Our proposed model directly puts fairness constraints on the predictions.

3. PRELIMINARIES

In this section, we discuss the formalization of individual fairness.

3.1 Individual Fairness

Cynthia et al. [8] introduces the concept of individual fairness, which is based on the idea that similar individuals should be treated similarly. This definition requires a similarity metric measuring the similarity between two individuals. Given two individuals x_i and x_j , a classifier H is individually fair if the difference of the predictions between the individuals are upper bounded by their dissimilarity. The definition is as following

$$D(H(x_i), H(x_j)) < d(x_i, x_j) \quad (1)$$

where D is the distance measure between the outputs of the classifier and d is the distance metric between the two individuals. The drawback of this definition is that a similarity metric is required. A similarity metric guaranteeing fairness is problem specific and requires strong assumptions, which obstructs its adoption [5].

3.2 Metric Free Individual Fairness

Hu et al. [19] proposed metric free individual fairness based on the idea that the qualification of an individual should not be influenced by his/her sensitive attribute. Thus, changing an individual's sensitive attribute should not change the prediction of a classifier. The definition of metric free individual fairness is following

$$D(P(Y|x_i, S = s_i), P(Y|x_i, S \neq s_i)) < \epsilon \quad (2)$$

where s_i is the sensitive attribute of individual i , D is the distance measure of the predictions, ϵ is an arbitrarily small

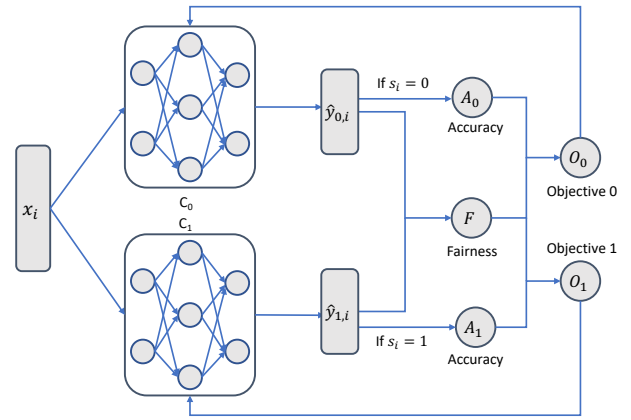


Figure 3: The architecture of the proposed model. The model consists of two classifiers C_0 and C_1 corresponding to sensitive attribute 0 and 1. An input vector x_i is fed into the two classifiers and the outputs are used to compute accuracy and fairness score. Note that if the sensitive attribute s_i is 0, accuracy A_0 and fairness F are combined to compute objective O_0 and only classifier C_0 is updated; otherwise, A_1 and fairness F are combined to form objective O_1 and classifier C_1 is updated.

positive number. This definition eliminates the requirement of a similarity measure between individuals. In this work, we develop a fair model based on this definition.

4. METHODS

4.1 Problem Statement

In this work, we focus on the task of identifying at-risk students. Given a student i with $((x_i, s_i), y_i)$, $x_i \in \mathbb{R}^P$ encodes the student's grades in courses taken prior to the target course; $s_i \in \{0, 1\}$ is the student's sensitive attribute such as gender or race; $y_i \in \{0, 1\}$ is the ground truth label indicating whether a student is at-risk (1) or not (0). We focus on a binary sensitive attribute, though our method can be easily extend to scenarios where the sensitive attribute is n-ary. We want to build a classifier to predict if a student will underperform in a future target course. The classifier needs to satisfy two constraints: 1) make predictions as accurate as possible and 2) the output of the classifier is individually fair as specified by Equation 2.

The model is trained in a course-specific manner, namely, we train a model for each target course. Given a target course, we extract all the students who have taken it. The courses these students have taken prior to the target course are extracted as prior courses. The students' grades in the prior courses are extracted to form a matrix X and the students' grades in the target course are Y . Students' sensitive attributes are denoted as S . We train a course-specific model on (X, Y) to predict whether students who have not taken the target course will fail it or not. Note that sensitive attributes S are not used as features.

4.2 Proposed Algorithm

In this section, we present the proposed model, multiple cooperative classifier model (MCCM). Figure 3 shows the

architecture of the proposed model. The model is composed of two classifiers, each of which corresponds to a sensitive attribute, e.g., male or female. Given an individual $((x_i, s_i), y_i)$, the feature vector x_i is fed into the two classifiers. The output of the classifier corresponding to s_i is the individual's probability of being positive, while the output of the classifier corresponding to $1 - s_i$ is the individual's probability of being positive if his/her sensitive attribute is changed. Based on the assumption of metric free individual fairness, to be fair the difference between the outputs of the two classifiers should be ignorable. In this work, the difference is the KL-divergence of the two outputs. In addition to fairness, we also care about the accuracy of the classifier. Therefore, for student i , the objective function we seek to optimize is as following

$$L_i = -y_i \log \hat{p}_{s_i, i} - (1 - y_i) \log(1 - \hat{p}_{s_i, i}) + \lambda \text{KL}(\hat{p}_{s_i, i}, \hat{p}_{1-s_i, i}) \quad (3)$$

where λ is a hyperparameter trading off between accuracy and fairness, $\hat{p}_{s_i, i}$ is the probability of being positive predicted by classifier s_i and $\hat{p}_{1-s_i, i}$ is the probability predicted by classifier $1 - s_i$. Note that, for L_i only the classifier corresponding to s_i is updated. The classifiers are feed-forward neural networks with two hidden layers. The activation function is chosen to be ReLU [12]. Dropout [16] is used to prevent overfitting.

Algorithm 1: Multiple Cooperative Classifier Model

Input : Data $D = \{((x_i, s_i), y_i)\}_{i=1}^N$, learning rate α , λ , number of iterations T , classifier C_0 and C_1 .

```

1 Initialize parameters  $\{\theta_0^0, \theta_1^0\}$ 
2 for  $t = 1, \dots, T$  do
3   Sample example  $((x_i, s_i), y_i)$  from  $D$ 
4   Feed  $x_i$  into classifier  $C_{s_i}$  and  $C_{1-s_i}$ 
5   Compute the loss  $L_i$  according to equation 3
6    $\theta_{s_i}^{t+1} = \theta_{s_i}^t + \alpha \frac{\partial L_i}{\partial \theta_{s_i}^t}$ 
7 return  $\{\theta_0^T, \theta_1^T\}$ 

```

5. EXPERIMENTAL PROTOCOL

5.1 Datasets

To evaluate the proposed model, we collect ten-year data at George Mason University from Fall 2009 to Fall 2019. We choose top five majors including Biology (BIOL), Civil Engineering (CEIE), Computer Science (CS), Electrical Engineering (ECE) and Psychology (PSYC). We only choose a course if there are at least 300 students who have taken it. We use a student's grade in prior courses to predict whether a student is at-risk of failing a target course. While preprocessing the data, we exclude courses that are not relevant to a major such as elective courses. Table 1 shows statistics of the data. From the table, we can see clear gender difference for different majors. Female students tend to choose Biology and Psychology majors, while male students are more prone to engineering majors such as Civil Engineering, Computer Science and Electrical Engineering. Overall, the proportion of African-American students is relatively small, especially for Civil Engineering and Computer Science.

We build course specific models, namely, for a target course we train a classifier to predict whether a student will fail

that course in the future. We define as at-risk student if the student's grade is lower than 3.0. Given a target course, the data related to that course is split into 75%, 15%, 15% for training, validation and testing, respectively.

5.2 Baselines

As in this work we focus on individual fairness, we compare our proposed model with several individually fair algorithms.

5.2.1 Logistic Regression (LR)

This baseline does not have a fairness constraint. It directly predicts if a student is at-risk or not. The input is a feature vector encoding a student's grades in prior courses. The output is the student's probability of failing the target course.

5.2.2 Rawlsian Fairness (Rawlsian)

The concept of Rawlsian fairness is that a worse candidate should never be favored over a better one. Joseph et al. [21] proposed an individually fair algorithm utilizing a contextual bandits as building block to implement Rawlsian fairness.

5.2.3 Learning Fair Representation (LFR)

The unfairness of a prediction comes from the correlation of the output with the sensitive attribute. Zemel et al. [32] proposed to remove the correlation by learning an intermediate representation and train a classifier on it.

5.2.4 Adversarial Learned Fair Representation (ALFR)

Edwards et al. [9] propose to remove sensitive information from representation by adversarial learning. An encoder maps the original feature vector to a latent embedding vector, from which an adversary tries to predict the sensitive attribute. While the adversary tries to predict the sensitive attribute, the encoder seeks to generate a representation that prevent the encoder from predicting it.

5.3 Evaluation Metrics

To evaluate if the proposed algorithm satisfy the accuracy and fairness constraints, we utilize three evaluation metrics **accuracy**, **discrimination** and **consistency**.

The **accuracy** metric assesses the predictive accuracy of the model, defined as following

$$\text{acc} = \frac{\sum_{i=1}^N \mathbb{1}(y_i = \hat{y}_i)}{N} \quad (4)$$

where N is the number of examples, \hat{y}_i is the prediction and y_i is the ground truth label.

Discrimination measures the difference between the groups' rate of being predicted as positive, mathematically expressed as following

$$\text{discr} = \left| \frac{\sum_{i=1}^N \mathbb{1}(s_i = 0) * \hat{y}_i}{\sum_{i=1}^N \mathbb{1}(s_i = 0)} - \frac{\sum_{i=1}^N \mathbb{1}(s_i = 1) * \hat{y}_i}{\sum_{i=1}^N \mathbb{1}(s_i = 1)} \right| \quad (5)$$

Consistency compares the predicted results of an individual with his/her k -nearest neighbors. If the predicted results

Table 1: Dataset Statistics

Major	#S	#C	#G	#M	#F	#AA	#NAA
BIOL	6,127	16	124,716	1,927(31.45%)	4,200(68.55%)	759(12.39%)	5,368(87.61%)
CEIE	450	7	23,708	338(75.11%)	112(24.89%)	27(6.00%)	423(94.00%)
CS	2,430	11	90,819	1,942(79.92%)	488(20.08%)	157(6.46%)	2,273(93.54%)
ECE	671	10	65,396	575(85.69%)	96(14.31%)	66(9.84%)	605(90.16%)
PSYC	5,110	17	84,504	1,200(23.48%)	3,910(76.52%)	694(13.58%)	4,416(86.42%)

#S total number of students, #C number of courses for prediction, #G total number of grades
 #M number of male students, #F number of female students, #AA number of African-American students
 #NNA number of non-African-American students.

is close to the results of the neighbors, consistency is high and the algorithm is fair. Consistency is defined as following

$$\text{consist} = 1 - \sum_{i=1}^N \frac{\sum_{n=1}^K |\hat{y}_i - \sum_{j \in \text{kNN}(x_i)} \hat{y}_j|}{K} \quad (6)$$

where $\text{kNN}(x_i)$ is the k -nearest neighbors of individual i .

We use Gower similarity [13] to measure the similarity between individuals. Gower similarity is defined as

$$\text{Gower}(i, j) = \frac{\sum_{k=1}^N w_k S_{ijk}}{\sum_{k=1}^N w_k} \quad (7)$$

where N is the number of features and w_k is the weight of the k -th variable, in this paper the weights are set to one; S_{ijk} is the contribution by the k -th variable. If the k -th variable is continuous, S_{ijk} is defined as

$$S_{ijk} = 1 - \frac{|x_{ik} - x_{jk}|}{r_k} \quad (8)$$

where x_{ik} is the value of k -th feature of i and r_k is the range of values for the k -th variable. If the k -th variable is categorical, S_{ijk} is 1 if $x_{ik} = x_{jk}$ or 0, otherwise.

6. EXPERIMENTAL RESULTS

6.1 Results and Analysis

We train a classifier for each course in a major to predict if a student will fail that course. The predictions are evaluated by using accuracy, discrimination and consistency. The results are averaged across the courses in a major. Table 2 shows the experimental results with gender as sensitive attribute. From the table, we can see that the proposed model **MCCM** achieves the best performance in mitigating bias in terms of discrimination. It is able to achieve both group fairness and individual fairness, although, it is designed for achieving individual fairness. The reason is that group and individual fairness are highly correlated so that achieving one helps achieving the other.

The predictions from **LR** model is highly biased as there is no fairness constraint imposed on it, but it performs well with respect to predicting accuracy. On average, the discrimination of **LR** is 7.3%. Other methods achieve fairness at the cost of accuracy. It is interesting to see that **Rawlsian** is not able to remove bias and in some cases it leads to even more unfair predictions. **Rawlsian** is based on the idea that a worse candidate should never be favored over a better one, which is implemented by interval chaining that is a weak fairness constraint. We can also observe from the

table that different majors have different level of bias, e.g., Psychology has the least bias while Computer Science has the highest bias with respect to the predictions of **LR**. The experimental results with race as sensitive attribute is shown in Table 3. The results are similar to those with gender as sensitive attribute.

6.2 Fine-grained analysis of the bias

To have a fine-grained view of the bias, we look at the data and predictions at the course level. In this section, we analyze the bias embedded in the data and predictions from **LR** and the proposed model **MCCM**. Figure 4 shows the fine-grained results with gender as sensitive attribute. For Figure 4, the data bias is that the proportion of at-risk female students subtracts the proportion of at-risk male students. Positive bias means female students are more likely to be predicted as at-risk; otherwise male students are more likely to be predicted as at-risk. For the predictions from the models, the bias is the female students' average probability of being predicted as at-risk students subtract that of male students.

First of all, as stated in Section 1, the overall data such as overall GPA by gender shows that male is minority groups. However, when looking at the course level, different courses have different minority groups. Figure 4 shows that in some courses male students are less likely to be at-risk. This insights can be used to inform future fairness work in educational data mining that a course specific model is desirable, considering that different courses have different minority groups. From the figures, we can also observe that data and machine learning models might have different bias direction. For example, in Figure 4(a), for course C0 the data bias is against male while **LR** and **MCCM** is against female. In addition, data bias does not necessarily lead to predictive bias. For example in Figure 4, all the courses show data bias. However, a no-fairness-constraint classifier, e.g., logistic regression has fair predictions in many courses.

7. CONCLUSION AND FUTURE WORK

The concerns about bias and discrimination of machine learning models are rising with the increasing of their adoption. In educational setting, we observe bias from a real-world dataset and machine learning models without fairness constraints exhibit non-ignorable biased predictions. Machine learning models are intended to aid students with their learning. However, unfair treatment of students can undermine their learning and graduation. To mitigate discrimination in educational data mining, in this paper, we proposed a fair machine learning model satisfying metric free individual

Table 2: Experimental results with gender as sensitive attribute.

Method	BIOL			CEIE			CS			ECE			PSYC		
	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)
LR	0.7662 0.0613 0.8152			0.6761 0.0837 0.7451			0.6628 0.1007 0.7569			0.7545 0.0980 0.7655			0.7769 0.0192 0.9578		
Rawlsian	0.5889 0.0807 0.8120			0.6250 0.0866 0.7052			0.5582 0.0913 0.8301			0.6660 0.1498 0.7036			0.7559 0.0960 0.9396		
LFR	0.6470 0.0369 0.9691			0.6983 0.0518 0.9631			0.6004 0.0228 0.9463			0.7389 0.0273 0.9912			0.7898 0.0248 0.9865		
ALFR	0.6802 0.0202 0.9675			0.7062 0.0240 0.9855			0.6124 0.0134 0.9821			0.7465 0.0114 0.9783			0.7903 0.0125 0.9878		
MCCM	0.6774 0.0163 0.9401			0.6415 0.0165 0.9823			0.6180 0.0038 0.9562			0.7394 0.0061 0.9717			0.7868 0.0023 0.9958		

acc = accuracy, discr = discrimination, consist = consistency.
 ↑ means higher is better; ↓ means lower is better.

Table 3: Experimental results with race as sensitive attribute.

Method	BIOL			CEIE			CS			ECE			PSYC		
	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)	acc(↑) discr(↓) consist(↑)
LR	0.7662 0.1004 0.8152			0.6761 0.1411 0.7451			0.6628 0.1085 0.7569			0.7545 0.1238 0.7655			0.7769 0.0276 0.9578		
Rawlsian	0.5854 0.1129 0.7870			0.5849 0.3658 0.7349			0.5561 0.1857 0.8007			0.6999 0.1446 0.7416			0.7608 0.0776 0.9570		
LFR	0.6202 0.0569 0.9051			0.7099 0.1722 0.9701			0.6107 0.0599 0.9897			0.7441 0.0800 0.9852			0.7874 0.0172 0.9933		
ALFR	0.6850 0.0505 0.9504			0.7274 0.0862 0.9688			0.6129 0.0086 0.9715			0.7435 0.0384 0.9887			0.7898 0.0156 0.9882		
MCCM	0.6563 0.0198 0.9340			0.7138 0.0114 0.9828			0.5895 0.0303 0.9968			0.7133 0.0013 0.9986			0.7857 0.0021 0.9974		

acc = accuracy, discr = discrimination, consist = consistency.
 ↑ means higher is better; ↓ means lower is better.

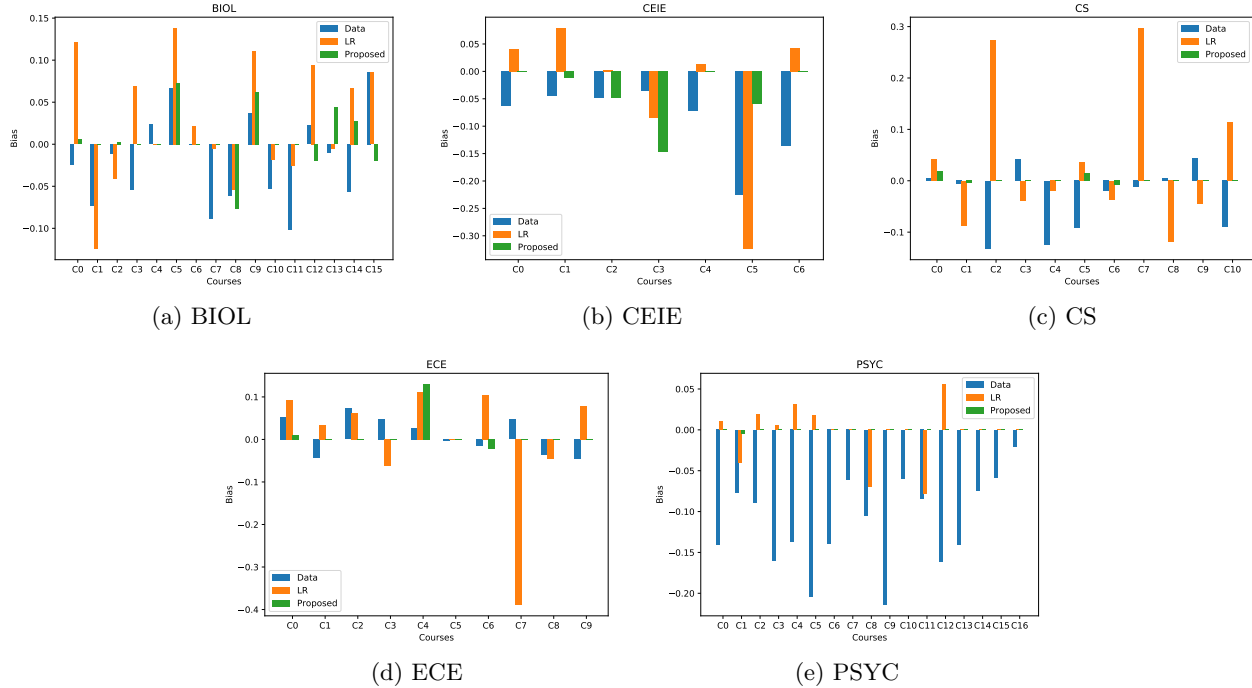


Figure 4: Bias of different courses with gender as sensitive attribute.

fairness. We evaluate the model’s performance on removing unfairness on datasets collected from an anonymous University. The results show the efficacy of the model on removing bias. Compared to other domains, educational data mining has its own characteristics. For example, in our dataset, when looking at university level, male and African-American students are biased against. However, at course level, different courses have different bias direction. This insights inform that future work on fairness in educational data mining should design course-specific models. In this work, we treat gender and race separately in terms of removing bias. In the future, we want to build models that treat gender and race as sensitive attributes simultaneously.

8. REFERENCES

- [1] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. *ProPublica*.
- [2] A. F. Botelho, R. S. Baker, and N. T. Heffernan. Improving sensor-free affect detection using deep learning. In *International Conference on Artificial Intelligence in Education*, pages 40–51. Springer, 2017.
- [3] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91, 2018.
- [4] Y. Chen, A. Johri, and H. Rangwala. Running out of stem: a comparative study across stem majors of

- college students at-risk of dropping out early. In *Proceedings of the 8th international conference on learning analytics and knowledge*, pages 270–279, 2018.
- [5] A. Chouldechova and A. Roth. The frontiers of fairness in machine learning. *CoRR*, abs/1810.08810, 2018.
- [6] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.
- [7] S. Doroudi and E. Brunskill. Fairer but not fair enough on the equitability of knowledge tracing. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 335–339, 2019.
- [8] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel. Fairness through awareness. *CoRR*, abs/1104.3913, 2011.
- [9] H. Edwards and A. Storkey. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.
- [10] A. Elbadrawy, A. Polyzou, Z. Ren, M. Sweeney, G. Karypis, and H. Rangwala. Predicting student performance using personalized analytics. *Computer*, 49(4):61–69, 2016.
- [11] M. Feldman, S. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact, 2014.
- [12] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.
- [13] J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.
- [14] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. *CoRR*, abs/1610.02413, 2016.
- [15] V. Hegde and P. Prageeth. Higher education student dropout prediction and analysis through educational data mining. In *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pages 694–699. IEEE, 2018.
- [16] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- [17] Q. Hu and H. Rangwala. Course-specific markovian models for grade prediction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 29–41. Springer, 2018.
- [18] Q. Hu and H. Rangwala. Academic performance estimation with attention-based graph convolutional networks. *arXiv preprint arXiv:2001.00632*, 2019.
- [19] Q. Hu and H. Rangwala. Cooperative contextual bandits for metric-free individual fairness. 2020.
- [20] S. Hutt, M. Gardener, D. Kamentz, A. L. Duckworth, and S. K. D’Mello. Prospectively predicting 4-year college graduation from student applications. In *Proceedings of the 8th international conference on learning analytics and knowledge*, pages 280–289, 2018.
- [21] M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth. Rawlsian fairness for machine learning. *arXiv preprint arXiv:1610.09559*, 1(2), 2016.
- [22] M. J. Kusner, J. Loftus, C. Russell, and R. Silva. Counterfactual fairness. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4066–4076. Curran Associates, Inc., 2017.
- [23] R. Morsomme and S. V. Alferez. Content-based course recommender system for liberal arts education. In *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, volume 748, page 753. ERIC, 2019.
- [24] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.
- [25] T. Ojha, G. L. Heileman, M. Martinez-Ramon, and A. Slim. Prediction of graduation delay based on student performance. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3454–3460. IEEE, 2017.
- [26] L. Paquette, J. Rowe, R. Baker, B. Mott, J. Lester, J. DeFalco, K. Brawner, R. Sottolare, and V. Georgoulas. Sensor-free or sensor-full: A comparison of data modalities in multi-channel affect detection. *International Educational Data Mining Society*, 2016.
- [27] A. Polyzou, N. Athanasios, and G. Karypis. Scholars walk: A markov chain framework for course recommendation. In *Proceedings of the 12th International Conference on Educational Data Mining*, pages 396–401, 2019.
- [28] A. Polyzou and G. Karypis. Feature extraction for classifying students based on their academic performance. *International Educational Data Mining Society*, 2018.
- [29] P. Sapiezynski, V. Kassarnig, and C. Wilson. Academic performance prediction in a gender-imbalanced environment. 2017.
- [30] T.-Y. Yang, R. S. Baker, C. Studer, N. Heffernan, and A. S. Lan. Active learning for student affect detection. In *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, pages 208–217. ERIC, 2019.
- [31] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *International conference on artificial intelligence in education*, pages 171–180. Springer, 2013.
- [32] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

Using online textbook and in-class poll data to predict in-class performance

Noah Hunt-Isaak
Dickinson College
noahhuntisaak@gmail.com

Peter Cherniavsky
George Mason University
pcherniavsky@gmail.com

Mark Snyder
George Mason University
msnyder14@gmu.edu

Huzefa Rangwala
George Mason University
rangwala@cs.gmu.edu

ABSTRACT

National failure rates seen in undergraduate introductory CS courses are quite high. In this paper, we develop a predictive model for student in-class performance in an introductory CS course. The model can serve as an early warning system, flagging struggling students who might benefit from additional support. We use a variety of features from the first few weeks of the course such as scores on assignments, interaction with the online textbook, and participation with the in-class polling system in order to train our models. We compare the performance of a number of machine learning algorithms on predicting final exam scores as well as final course grade. We find that the Support Vector Machine and AdaBoost are the most effective, and that we can achieve increasingly accurate predictions as we use data from further into the course. The regression coefficients give us insights into which features are most correlated with student success, suggesting that certain types of assignments are more indicative of learning than others.

Keywords

education data mining, performance prediction, early warning system

1. INTRODUCTION

An enduring challenge in higher education is student dropout. National studies [1] have reported a relatively stable average six-year graduation of approximately 60% over the past decade. The problem is acute in STEM (Science, Technology, Engineering and Mathematics) fields where a well-trained and educated workforce is essential for national growth and economy. As the volume and variety of data collected in both traditional and online university offerings continue to expand, educational data mining [1, 20] provides the promise to assist students and improve overall student retention.

Noah Hunt-Isaak, Peter Cherniavsky, Mark Snyder and Huzefa Rangwala "Using online text books and in-class quizzes to predict in class performance" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 438 - 443

Close to 30% of the students who enrolled in CS 112, the introductory CS course at George Mason University, during the Fall 17 and Fall 18 semesters failed the course. This is close to the national average pass rate in CS1 courses found by Watson and Li [23].

The focus of this paper is to develop a model for predicting student performance in a course that they are currently enrolled in based on performance in the first few weeks of the course. An accurate predictive model may serve as an early warning system which would alert the professor to students who are struggling, at which point additional support could be provided. Attempting to make predictions too early on in the course would result in inaccurate predictions, while waiting too long will mean it is too late to take any preventative action. Howard et al. suggest that the optimal time to employ such early warning systems is right around the midway point of the course, as this provides a good balance between reasonable predictive accuracy while still allowing time to make corrective changes [6].

In this paper we extract features that capture student behaviors such as consistency, dedication, and grit, and use these alongside the gradebook to make accurate predictions as early as possible. We investigate various machine learning (ML) regression algorithms including Ridge, Lasso, Elastic-Net, Support Vector Machine, AdaBoost, Gradient Boosting, Bagging, and Random Forest. Each algorithm is tuned by applying a grid search to the parameters space. We predict both final exam scores and final course scores, comparing the different models' prediction performance on both.

Finally, we examine the coefficient weights of the most successful models to determine which features are the most significant in making predictions. Discovering patterns which seem to help or hurt students could help the professor to better structure the course in the future, as well as discover more general trends which could be applied elsewhere.

2. PRIOR WORK

A number of attempts have been made at developing grade prediction models and early warning systems similar to what we wish to accomplish here. Most commonly, this has been done in the context of massive open online courses (MOOCs) such as in Ren et al. [19]. Li et al. [16] made an early

prediction model for a blended course. They had access to homework and test scores from the first 6 weeks of the course and used this data to predict students' final letter grades. However, they were only able to slightly improve upon a base model of predicting all A's (47.8% accuracy) by using an SVM (51.4% accuracy). This was probably a result of the course being too easy, which resulted in a fairly uniform data set. Elbadrawy et al. [11] looked at predicting performance on activities within a course, also using multi-regression models. They took into account a combination of features including demographic information, historical performance and course interaction data from the LMS. Another study done by Nam and Samson [17] investigated the impact of student behavioral signals in early warning system predictions. Costa et al. [4] investigated the effectiveness of four algorithms in predicting student failure in two computer science courses along with the effect of preprocessing data and fine tuning the algorithms on their performance. The preprocessing and fine tuning was found to improve performance for the most part. Ren et al. [18] developed Additive Latent Effect Models to predict student performance in a future course. It used factors from the student, course, and instructor for the prediction. The model performed better than all the baselines. Crossley et al.[5] did a study that used click-stream, language, and demographic data to predict the performance of elementary school students in a math course. The click-stream and language data was from an online math tutoring system. They found that the linguistic, click-stream, and demographic factors explained 14% of the variance in the math score and random factors of the student explained 30% of the variance in the math score. Elbadrawy et al. [10] developed a personalized linear multiple regression (PLMR) and other models to predict grades in a future course and grades in a future assignment within regular courses and a massive open online course (MOOC). Matrix factorization (MF) and PLMR outperformed traditional models in predicting student success in a future course and PLMR was useful in predicting grades within regular courses and a MOOC.

Several papers have covered the usage of zyBooks in CS courses, the online interactive textbook used in the course here. Most students complete the textbook [8, 7, 9]. The relationship between completion rate of the textbook and percentage of the grade that the textbook work is worth levels off at a certain point [8, 9]. The acquisition rate of zyBooks is higher than traditional textbooks [7]. Students mostly do not cheat when using zyBooks [8, 9]. Students' use of zyBooks is stable throughout the semester [7].

Unlike most of the prior studies which have been done in this area, we are exclusively using data from within the context of the course itself (i.e. we don't consider any student demographic or background information). We do, however, use a wider range of interaction data for students.

3. DATASET

Our data is taken from two sections of George Mason's introductory CS course, CS112. CS112 is taught in Python and covers a range of basic programming concepts including variables, conditionals, functions, loops, dictionaries, files, classes, and recursion. The courses were taught during the Fall semesters of 2017 and 2018, for an overall enrollment of 1,197 students. It includes each student's grade book from the semester, including scores on homework, labs, projects, and tests. The grade book also contains additional information such as whether or not the student was flagged for an honor code violation and how many of their allotted late submissions were used. CS112 was taught with an accompanying interactive online textbook, zyBooks, from which we have submission logs. Additionally an online polling system, Pytania [21], was used routinely during lectures. The final grade composition for the course was computed as follows:

Category	Percent	Notes
Projects	40%	drop 1 lowest
Labs	10%	drop 2 lowest
Pytania Particip.	2%	up to 1% bonus for correct answers
zyBook readings	3%	(drop 3 lowest-completion sub-sections)
Tests	20%	(10% each test)
Final Exam	25%	(must pass final to pass class)

Note that students were required to pass the final in order to pass the course, so that even if a student's raw score would have given them a D or above, they ended up with an F if they failed the final (this impacted 82 out of the 1,197 students, or about 6.85%). The overall grade distribution across both semesters of the course can be seen in Figure 1, which uses the following letter grade assignment to map raw scores to letter grades.

Grade	Score	Grade	Score	Grade	Score	Grade	Score
A+	98%	B+	88%	C+	78%	D	60%
A	92%	B	82%	C	72%	F	0%
A-	90%	B-	80%	C-	70%		

4. METHODS

All of the students who were flagged with an honor code violation were removed from the data set and not considered when making predictions. We consider their data as inaccurate records of effort, ability, and expected outcomes.

4.1 Feature Engineering

New features were engineered based on the Pytania and zyBook data in an attempt to capture behavioral patterns of students. In particular, we aimed to represent qualities such as participation, consistency, and grit, all of which may be important factors in predicting success. Note that the Pytania data is essentially a measure of attendance and in-class participation.

The Pytania data in its raw form consisted of rows corresponding to a single question answered by a single user with a timestamp and whether or not the question was answered correctly. We needed to extract from this a set of features for

each user which could be fed into our predictive model. The first attempt was to simply add two new features for each student, the number of Pytania questions attempted and the number answered correctly. However we could capture more information from the data by also taking into account the chronology of when students were answering questions. For example, a student who missed the first week but has answered every question afterwards ought to be more highly considered than one who answered everything for the first couple of weeks but has since dropped off. To capture this information, binning was used to create multiple new features dependent on the number of questions answered within a certain time frame. In particular, a feature is added for the number of question answered in each 1 week period through the semester. For example, there is a bin for the most recent week, and another for the week before that, and so on until the start of the semester. In this way we expect the model to weight more recent bins more highly as they are more likely to predict future performance compared to using past performance as a predictor.

We also wanted to have some measure of consistency of student participation. This was measured by taking the standard deviation of the weekly bins. The result was added as a new feature to the model. Students with a high amount of inconsistency would perhaps be expected to perform worse, or at least be more difficult to predict accurately.

The zyBook data was similar in form to the Pytania data, recording individual attempts at a question by a student. Unlike with the Pytania data, students could have multiple attempts at each question. Across all users, the average number of submissions per question is 1.5. The features extracted from this data include total number of attempts (submissions), total number of correct submissions (max one per problem), and average earliness (measured as the difference in time between the problem due date and the first correct submission). The correct submissions feature is divided up by chapters from the textbook, and special ‘challenge’ problems, which are more difficult and involved, are handled separately from regular ‘participation’ problems.

4.2 Normalization

Normalization was applied to ensure that no one feature had too dominant of an impact. Each column of features was mapped to the range 0-1 such that the max observed value became a 1 and the min value became a 0. This had a minimal positive effect on results.

4.3 Algorithms

After the features were generated, we ran experiments to determine which algorithm does the best job at predicting student performance. We tested a wide variety of regression algorithms in our experiments including Linear, ElasticNet, Lasso, and Ridge regression models [24, 22, 15, 3]. We also used the support vector regression algorithm (SVR). Finally, we used a number of ensemble algorithms including AdaBoost, Bagging, Gradient Boost and a Random Forest approach [12, 2, 13, 14]. We employed the Python module `sk.learn` for implementations of each algorithm and each algorithm was additionally tuned for optimal parameters using GridSearch (the final parameters can be seen in appendix A).

4.4 Metrics

Two relevant outcomes for prediction were examined: predicting the final RAW score in the course and predicting the score on the final exam (test 3, “T3”). Final RAW score is arguably more important to predict as this is what impacts the student’s GPA and is the most commonly used metric for success in a course. However, many of the features being used to predict RAW score are also directly correlated with RAW scores (e.g. projects make up 40% of the final course grade). This makes the prediction less meaningful as the regression weights may come to simply recreate the exact course weighting for assignment. Hence we also predict scores for the final exam, a distinct assessment which is not directly calculated from any of the other features. The final exam is also arguably a better metric for how much of the content a student was able to truly learn.

For regression tasks such as predicting the RAW score we used mean absolute error (MAE), root mean squared error (RMSE), and the coefficient of determination (R2) score as our metrics of evaluation (as defined in figure 2). Note that MAE and RMSE are in the range $[0, \infty)$, and lower scores are better. The RMSE is a commonly used metric for evaluation of regression models, which punishes big errors due to the squared error term. The MAE on the other hand weights all individual differences equally on a linear scale. The result is a more interpretable score, which represents the average error in our predictions. (e.g. a MAE of 10 while predicting final exam scores would mean that the mean prediction was 10 points off from guessing the true value). R2 is in the range $(-\infty, 1]$, where higher scores are better. The R2 score compares the effectiveness of a model to a simple baseline which predicts the mean value for each instance. A score of 0 implies a model which makes no improvement upon this baseline, while a negative score implies a model which is worse than the baseline. A score of 1 perfectly predicts each true value.

5. EXPERIMENTAL RESULTS

Figure 3 shows the resulting MAE, RMSE, and R2 scores for predicting final exam and raw scores using the first 9 weeks of data in a 16 week course. We see that the AdaBoost regressor achieved the lowest MAE score when predicting final exam scores, with an average error of 8.02. The SVR and Bagging regressors also performed well here. For predicting the raw course score, the Ridge regressor performed the best overall, with the top RMSE of 7.03, and an R2 score of 0.84. The Lasso regressor outperformed Ridge slightly in terms of MAE where it achieved a score of 5.42. For the sake of comparison, we also present a ‘baseline’ score which was calculated simply by predicting the mean value for each student (note that this baseline will have an R2 score of 0 by definition).

Using the optimal algorithms for each category from above, we ran experiments to determine how accurately we can make predictions at various points throughout the semester. In particular, we used the AdaBoost regressor for predicting final exam score and the Ridge regressor for predicting final RAW score. Figure 4 shows the results for this experiment, plotting MAE scores as a function of the number of weeks of data used. As expected, the accuracy of the prediction improves as more data is available for use through the first

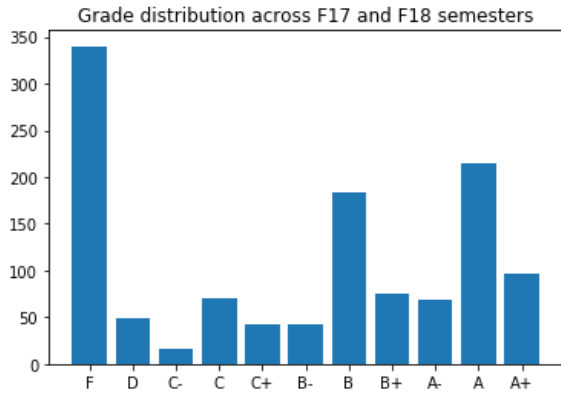


Figure 1: CS112 Grade Distribution

MAE	$\frac{1}{n} \sum y_{true} - y_{pred} $
RMSE	$\sqrt{\frac{1}{n} \sum y_{true} - y_{pred} ^2}$
R^2	$1 - \frac{\sum (y_{true} - y_{pred})^2}{\sum ((y_{true} - y_{true\ mean})^2)}$

Figure 2: Evaluation Metrics

(where y_{pred} and y_{true} correspond to the predicted value and the actual value respectively of a particular data point, n is the number of samples, and summations go over all n samples)

	Predict Final Exam			Predict RAW		
	MAE	RMSE	R2	MAE	RMSE	R2
Baseline	15.12	22.48	0.0	13.07	18.25	0.0
LinearRegression	9.98	13.51	0.47	5.72	7.55	0.82
Ridge	8.86	13.11	0.47	5.68	7.03	0.84
Lasso	9.95	13.11	0.50	5.42	7.07	0.84
ElasticNet	9.76	12.96	0.51	5.77	7.37	0.83
SVR	8.31	12.47	0.55	5.57	7.63	0.81
AdaBoost	8.02	10.97	0.65	6.11	9.20	0.73
GradientBoosting	9.32	14.11	0.43	7.97	12.76	0.49
Bagging	8.76	11.76	0.60	5.91	7.96	0.80
RandomForest	9.89	14.46	0.40	7.83	12.12	0.54

Figure 3: Regression Algorithm Comparison

10 weeks of the semester. We also see that predictions of raw score are consistently better than predictions of final exam score. This makes sense as the raw score is in fact calculated directly from some of the features we are using in our predictive model, while the final exam score is not.

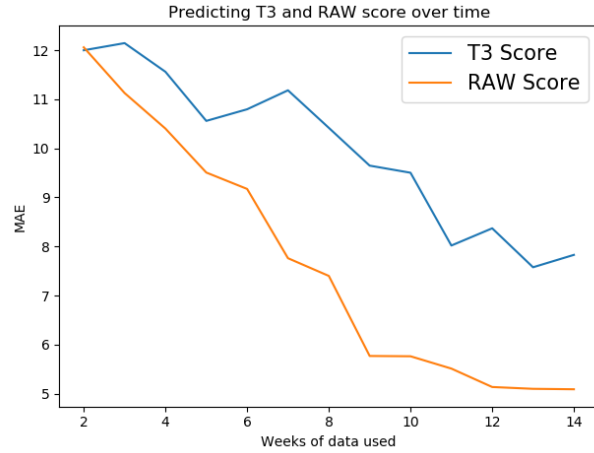


Figure 4: Predicting T3 score over time

Figures 6 and 7 show the values of the coefficient weights for each feature from the optimal final exam and raw score models run using 8 weeks of data (see figure 5 for coefficient definitions). For final exam prediction, it is not surprising to see that performance on the first exam, T1, is the best predictor. Next, other substantial graded assignments such as projects 2 and 3 and lab 5 also have fairly high weights. The Pytania features overall have small weights, suggesting that participation in the polling system is not strongly correlated with success. It may be the case that students did not take Pytania participation very seriously as it only accounted for 2% of their semester grade. However, we do observe that pwa:0 and pwa:1, which correspond to questions answered in the most recent weekly periods, are more significant than the earlier weeks, meaning that recent activity is a better indication of a student doing well than activity early on in the course. For the zyBook features we see that each chapter has a small positive weight. The strongest positive zyBook weights are cha_lacp (the number of challenge problems successfully completed) and zy_early (the average earliness of submissions). Interestingly, zy_attempt (corresponding to the total number of attempted submissions) and extra_sub

Feature Name	Meaning
L1E	Lab 1 Exercise
L4T	Lab 4 Test
L6Q	Lab 6 Quiz
P1	Project 1
T1	Test 1
cha_lacp	Challenge zyBook problems successfully submitted
extra_sub	Resubmissions to a previously correctly submitted problem
pytania_std	Standard deviation across Pytania bins
pwa:1	Pytania participation last week
pwa:2	Pytania participation two weeks ago
zychp1	zyBook correct submissions from chapter 1
zy_attempt	Total count of zyBook submissions
zy_early	Average earliness of zyBook submissions

Figure 5: Regression Coefficients

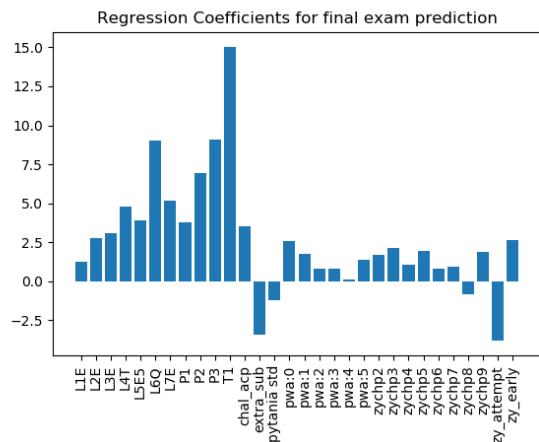


Figure 6: Regression coefficients T3

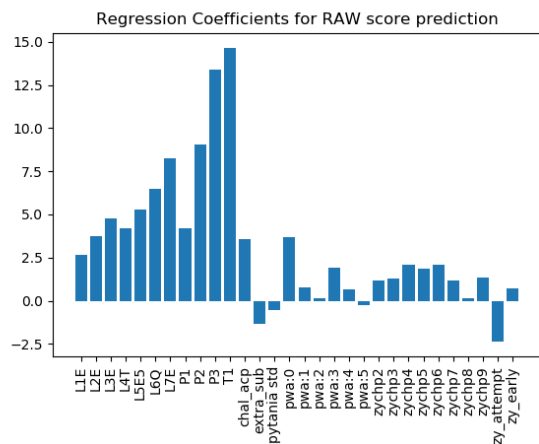


Figure 7: Regression coefficients Raw

(corresponding to the total number of additional submissions to a problem which had already been completed) both have negative weightings. The first could be explained by the fact that students who struggle and require multiple submissions to a problem are likely weaker students. The second is more difficult to account for, as one might expect subsequent attempts at a problem for which the student already has credit to be a sign of determination to learn the material.

6. DISCUSSION

In this paper we formulated an in-class predictive model for student performance in a CS1 course taught at George Mason University. Our model used a feature set solely derived from in-class activities and assignments rather than relying on past information or demographics. We employed a variety of machine learning algorithms and tested their accuracy as a function of the number of weeks of data used from the semester. Our results show that both final raw scores and final exam scores can be predicted with a high level of accuracy as early as 6 or 7 weeks into the course. Thus it could be effectively employed as an early warning system, such that students could have a good sense of what grade they

will end up with if they remain on their current trajectory.

We note that often in cases where multiple features had equivalent direct grade contributions (due to course weighting) there was a discrepancy in the predictive power amongst these features. For example, we found that recent participation was a more relevant predictor of ultimate success than past participation, even though it is equivalent in terms of direct impact on a student's grade. Similarly, we found that engagement with certain textbook chapters, and success on certain labs and quizzes are more indicative of success than others. Thus there could be a situation in which two students have the same class grade, and yet one is flagged as a struggling student while the other is not. Herein lies the value of using such a model; it can discover nuances and patterns in student performance which an instructor (particularly in an introductory course with many students) would otherwise be unable to detect.

6.1 Future Work

In this work we considered only in-class information. It would be interesting to see how much improvement could be made to the model by also considering past information such as prior course scores and demographics. It would also be worth trying to make predictions of other CS courses, or non-CS courses, and comparing the accuracy of predictions as well as which features stand out as strong predictors.

We propose that our predictive model could serve as an early warning system, triggering intervention. However, we do not suggest exactly how or when this intervention should take place. Studies would have to be performed on different forms of intervention to determine which methods work best.

7. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation REU grant for Education Data Mining at George Mason University, award 1757064. We would also like to thank Dr. Rangwala (PI of the REU site) for his contributions to our work and his invaluable guidance throughout the research process, as well our fellow REU student participants for their continual support and encouragement.

8. REFERENCES

- [1] Ryan Shaun Baker and Paul Salvador Inventado. Educational data mining and learning analytics. In *Learning analytics*, pages 61–75. Springer, 2014.
- [2] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.
- [3] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [4] Evandro B Costa, Balduino Fonseca, Marcelo Almeida Santana, Fabrícia Ferreira de Araújo, and Joilson Rego. Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *Computers in Human Behavior*, 73:247–256, 2017.
- [5] Scott Crossley, Shamyia Karumbaiah, Jaclyn Ocumpaugh, Matthew J Labrum, and Ryan S Baker. Predicting math success in an online tutoring system using language data and click-stream variables: A longitudinal analysis. In *2nd Conference on Language*,

Data and Knowledge (LDK 2019). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

- [6] M. Meehan E. Howard and A. Parnell. Contrasting prediction methods for early warning systems at undergraduate level. *Internet Higher Educ.*, pages 66–75, 2018.
- [7] Alex Edgcomb, Daniel de Haas, Roman Lysecky, and Frank Vahid. Student usage and behavioral patterns with online interactive textbook materials. In *International Conference of Education, Research and Innovation (ICERI), Spain*, 2015.
- [8] Alex Edgcomb, Frank Vahid, Roman Lysecky, and Susan Lysecky. Getting students to earnestly do reading, studying, and homework in an introductory programming class. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, pages 171–176. ACM, 2017.
- [9] Alex Daniel Edgcomb, Frank Vahid, Roman Lysecky, and Susan Lysecky. An analysis of incorporating small coding exercises as homework in introductory programming courses. In *ASEE Annual Conference and Exposition, Conference Proceedings*, volume 2017, 2017.
- [10] Asmaa Elbadrawy, Agoritsa Polyzou, Zhiyun Ren, Mackenzie Sweeney, George Karypis, and Huzefa Rangwala. Predicting student performance using personalized analytics. *Computer*, 49(4):61–69, 2016.
- [11] Asmaa Elbadrawy, Scott Studham, and George Karypis. Personalized multi-regression models for predicting students performance in course activities. *UMN CS*, pages 14–011, 2014.
- [12] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.
- [13] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.
- [14] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.
- [15] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- [16] Hengxuan Li, Collin F Lynch, and Tiffany Barnes. Early prediction of course grades: Models and feature selection. *arXiv preprint arXiv:1812.00843*, 2018.
- [17] SungJin Nam and Perry Samson. Integrating students’ behavioral signals and academic profiles in early warning system. In *International Conference on Artificial Intelligence in Education*, pages 345–357. Springer, 2019.
- [18] Zhiyun Ren, Xia Ning, and Huzefa Rangwala. Ale: Additive latent effect models for grade prediction. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 477–485. SIAM, 2018.
- [19] Zhiyun Ren, Huzefa Rangwala, and Aditya Johri. Predicting performance on mooc assessments using multi-regression models. *arXiv preprint arXiv:1605.02269*, 2016.
- [20] Cristobal Romero and Sebastian Ventura. Educational data mining and learning analytics: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1355, 2011.
- [21] M Snyder. Large-lecture participation via (free) pytanita student response system. In *Poster presented at Innovations in Teaching and Learning 2018, Fairfax, VA.*, 2018.
- [22] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.
- [23] Christopher Watson and Frederick W.B. Li. Failure rates in introductory programming revisited. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education, ITiCSE ’14*, pages 39–44, New York, NY, USA, 2014. ACM.
- [24] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

APPENDIX

Appendices

A. GRIDSEARCH PARAMETERS

Grid search found the optimal combination of parameters for each machine learning algorithm. The following parameters were used (any unlisted used the default value from sklearn):

Ridge: alpha=1.0, solver=‘lsqr’

Lasso: alpha=0.1

ElasticNet: alpha=0.1, l1_ratio=0.9

SVR: C=1, gamma=0.001, kernel=‘linear’

AdaBoost: learning_rate=0.2, loss=‘exponential’, n_estimators=80

GradientBoost: criterion=‘mae’, learning_rate=0.01, loss=‘huber’, max_depth=None, min_samples_leaf=1, min_samples_split=10, n_estimators=80, subsample=0.8

Bagging: bootstrap_features=True, max_features=20, n_estimators=80

RandomForest: criterion=‘mae’, min_samples_leaf=0.1, min_samples_split=0.1, n_estimators=40

Online Academic Course Performance Prediction using Relational Graph Convolutional Neural Network

Hamid Karimi^{1*}, Tyler Derr^{1*}, Jiangtao Huang², Jiliang Tang¹
¹ Michigan state University, {karimiha, derrytle, tangjili}@msu.edu
² Nanning Normal University, China, hjt@gxtc.edu.cn

ABSTRACT

Online learning has attracted a large number of participants and is increasingly becoming very popular. However, the completion rates for online learning are notoriously low. Further, unlike traditional education systems, teachers, if any, are unable to comprehensively evaluate the learning gain of each student through the online learning platform. Hence, we need to have an effective framework for evaluating students' performance in online education systems and to predict their expected outcomes and associated early failures. To this end, we introduce Deep Online Performance Evaluation (DOPE), which first models the student course relations in an online system as a knowledge graph, then utilizes an advanced graph neural network to extract course and student embeddings, harnesses a recurrent neural network to encode the system's temporal student behavioral data, and ultimately predicts a student's performance in a given course. Comprehensive experiments on six online courses verify the effectiveness of DOPE across multiple settings against representative baseline methods. Furthermore, we perform ablation feature analysis on the student behavioral features to better understand the inner workings of DOPE. The code and data are available from <https://github.com/hamidkarimi/dope>.

Keywords

Online courses, Student behavior modeling, Knowledge graph, MOOC, Graph neural networks

1. INTRODUCTION

Online learning has higher dropout and failure rates than traditional education systems. For instance, the completion rates of Massive Open Online Courses (MOOCs), an extension of online learning technologies, are low (0.7%-52.1%, with a median value of 12.6%, reported by [20]). We also see similar situations in other online courses from universities such as Open University in the UK and China [19].

*Equal contribution and co-first author

Hamid Karimi, Tyler Derr, Jiangtao Huang and Jiliang Tang "Online Academic Course Performance Prediction using Relational Graph Convolutional Neural Network" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 444 - 450

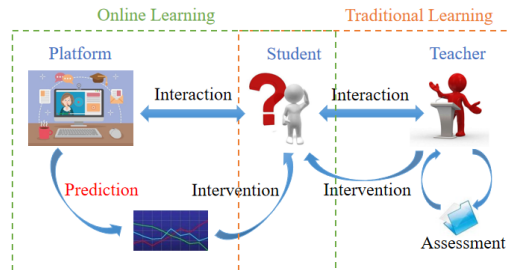


Figure 1: Visual comparison of the learning/intervention process between online and in-person education systems.

Furthermore, since students typically drop out early in the courses [33], the platform is desired to detect which student is likely to drop out (or fail) as early as possible to intervene and to hopefully prevent these negative outcomes. Then the question is how we can assess students' performance and detect those who are likely to drop out or fail in an online course. To answer this question, we first need to take a closer look at the online learning system and see how it differs from traditional learning.

As illustrated in Figure 1 (right side), in the traditional learning setting, instructors can interact with students, assess their performance, and take action to provide intervention if they sense a student is likely to perform poorly in the class. In online learning systems, however, the students primarily interact with the online platform, so we face a setting depicted in the left side of Figure 1. In this setting, there is inherently less interaction between students and instructors. More specifically, due to the high student-teacher ratio, teachers, if any, in the online learning systems are unable to comprehensively evaluate the learning gain of each student. Thus, we seek to develop a methodology that can harness the interactions of students with an online platform and accurately predict the course outcome (e.g., *pass or fail*). Such a system could then be used in real-time throughout the course to identify the students who are predicted to perform poorly and provide some intervention to them with the limited resources that are inherent in online systems.

Given the above discussion, we propose a framework named Deep Online Performance Evaluation (DOPE) to predict students' course performance in online learning. DOPE first models the student course relations of the online system as a knowledge graph. To incorporate an aggregated overview of

the students and courses in the online system, DOPE learns student and course embeddings from our knowledge graph. More specifically, we employ a relational graph neural network [34] that can handle the rich attribute information found in our knowledge graph (e.g., student demographic data). Then, our proposed approach utilizes a recurrent neural network (RNN) to encode the temporal student behavioral data into some features. More specifically, the student behavioral data is coming from student click patterns extracted and aggregated into weekly snapshots that represent how they have interacted with the online learning system. Finally, the student and course embeddings (extracted from the knowledge graph) are combined with the encoded behavioral data extracted for the given student and course and are fed to a classifier to predict a student's performance. In summary, our contributions are as follows.

1. We propose the use of a knowledge graph to model complex online learning environments to allow more rich data to be extracted as compared to representing the data in a traditional unstructured way; and
2. Our proposed framework to predict student course outcomes contains two novel components, namely a relational graph neural network to extract student and course embeddings from the formed knowledge graph and a recurrent neural network model for encoding student behavioral data according to their clicks in the online system.

2. PROBLEM STATEMENT

Suppose from the set of courses in an online system we have a subset of m courses denoted as $\mathcal{C} = \{c_1, c_2, \dots, c_m\}$. Furthermore, let there be n students having enrolled in at least one of the m courses in \mathcal{C} , which we denote as $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$. For each course c_j , we assume there are some course features that can be represented as the vector $\mathbf{f}_j \in \mathbb{R}^{d_c}$ with d_c being the dimension size after encoding the course features. Similarly for each of the students s_i we assume there has been some collected demographic information that can be represented as the vector $\mathbf{d}_i \in \mathbb{R}^{d_s}$ with d_s being the dimension size after having encoded the student demographic data. In addition to the demographic data, the system is assumed to have collected some sequential behavioral data for each student s_i enrolled in course c_j that we represent as $\mathbf{B}_{ij} = [\mathbf{B}_{ij}^1, \mathbf{B}_{ij}^2, \dots, \mathbf{B}_{ij}^k]$ where $\mathbf{B}_{ij}^w \in \mathbb{R}^q$ represents an encoding of the behavior for student s_i during the w^{th} week of course c_j , k represents the number of weeks for which behavioral data was collected, and q is the dimension of the encoded weekly student behavior. In other words, we have a tensor of student behavioral data $\mathbf{B} \in \mathbb{R}^{n \times m \times k \times q}$. For each student s_i , we represent their performance outcome in course c_j as o_{ij} , where we assume there can be P outcomes (denoted by the set $\mathcal{P} \in \mathcal{P}$).

Now, given the notations listed above, we seek to learn a model $f(\cdot|\theta)$ having parameters θ such that it can predict the course student outcomes \mathcal{O} as follows:

$$M(\mathcal{C}, \mathcal{S}, \mathcal{F}, \mathcal{D}, \mathbf{B}, \mathcal{O}, f(\cdot|\theta)) \rightarrow \hat{\theta}$$

where we use M to denote the machine learning (artificial intelligence) process, \mathbf{B} is used to represent the behavioral (e.g., click) data for a given set of courses \mathcal{C} using only the first k weeks of data, \mathcal{F} represents the set of course features of \mathcal{C} , \mathcal{D} denotes the set of demographic data for the students

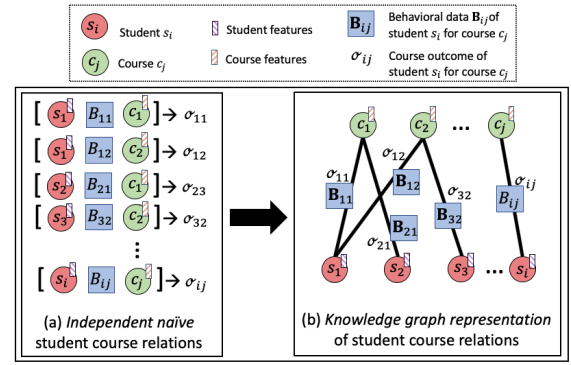


Figure 2: Visualizing the traditional representation used in prior supervised learning prediction models as compared to our knowledge graph representation.

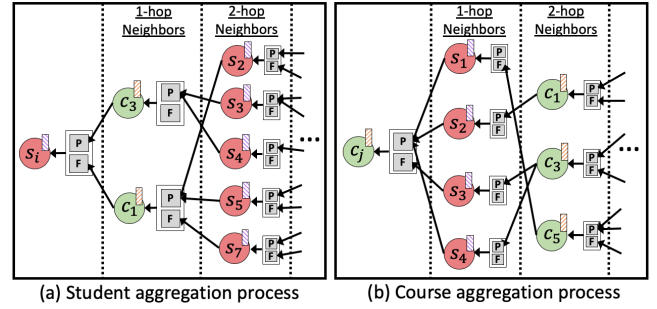


Figure 3: Visualizing the aggregation process in how both a student and course embedding are formed from their knowledge graph multi-hop neighborhood.

in \mathcal{S} , \mathcal{O} represents the performance outcomes of the students in \mathcal{S} and the learned parameters of $f(\cdot|\theta)$ are given by $\hat{\theta}$.

3. PROPOSED MODEL

In this section, we explain our proposed model in detail.

3.1 Knowledge Graph Representation

We first model the historical online course data in the form of a knowledge graph, as shown in Figure 2. Our knowledge graph formulation in Figure 2(b) offers a richer representation than a traditional independent naive student course relation representation shown in Figure 2(a). This is because through this graph structure we can leverage the relations between students and courses beyond that seen in Figure 2(a). We let $\mathcal{G} = \{\mathcal{C}, \mathcal{S}, \mathbf{X}_c, \mathbf{X}_s, \mathbf{B}, \mathbf{A}\}$ represent a knowledge graph \mathcal{G} containing the set of m course nodes \mathcal{C} , set of n student nodes \mathcal{S} , course features $\mathbf{X}_c \in \mathbb{R}^{n \times d_c}$ constructed from \mathcal{F} , student demographic features $\mathbf{X}_s \in \mathbb{R}^{m \times d_s}$ constructed from \mathcal{D} , the behavioral data \mathbf{B} representing complex sequential edge features, and an adjacency tensor $\mathbf{A} \in \mathbb{R}^{n \times m \times P}$ constructed from the P different student-course outcome relations where $\mathbf{A}_{ij}^p = 1$ if $o_{ij} \in \mathcal{O}$ and $o_{ij} = p$ (with $\mathbf{A}_{ij}^p = 0$ otherwise). Now, given the knowledge graph G , we seek to extract student and course embeddings by using a relational graph neural network.

3.2 Relational Graph Neural Network

Recently, graph neural networks (GNNs) [38, 39] have become increasingly popular due to their ability to utilize deep

learning on graph structure data. One popular class of GNNs is the graph convolutional networks (GCNs) [5, 27, 8, 7], which are constructed with roots from the classical CNNs. The general idea of these GCN models is that we would like to learn a better set of latent features. In the context of our problem, to better understand and represent a student, rather than directly using their features alone, we could use a 1-layer GCN that would incorporate the features of all the courses that the student has taken. For example, in Figure 3(a), the 1-hop neighbors would be utilized in a 1-layer GCN model taking into consideration the course c_3 that they passed and c_1 that they failed. Then, it is natural to see in Figure 3(a) that using a 2-layer GCN would further incorporate the 2-hop neighbors which would include information from all the classmates of s_i for each of the two courses they have taken, and thus providing further context into learning a more comprehensive embedding for student s_i . We specifically harness the ability of a relational graph convolutional network [34]. Next we will provide the details on how the first layer (or equivalently a 1-layer) GCN is able to construct learned representations $\mathbf{h}_{s_i}^1$ and $\mathbf{h}_{c_j}^1$ for the student s_i and course c_j , respectively, from the initial student features \mathbf{X}_s , course features \mathbf{X}_c , and adjacency tensor \mathbf{A} in our knowledge graph representation.

–**First Layer Embeddings.** First, we recall that connections between students and courses are stored in the tensor \mathbf{A} where $\mathbf{A}_{ij}^p = 1$ if $o_{ij} \in \mathcal{O}$ and $o_{ij} = p$ (with $\mathbf{A}_{ij}^p = 0$ otherwise). Thus, we define for a student s_i their set of courses for which they had outcome p as $\mathcal{N}_s^p(s_i)$. Similarly, we define for a course c_j their set of students that received the outcome p as $\mathcal{N}_c^p(c_j)$. Now, given these new notations, we can define the first layer representations $\mathbf{h}_{s_i}^1$ and $\mathbf{h}_{c_j}^1$ for the student s_i and course c_j , respectively, as follows:

$$\mathbf{h}_{s_i}^1 = \sigma \left(\mathbf{W}_{self}^1 \mathbf{X}_{s[i]} + \sum_{p \in \mathcal{P}} \frac{1}{|\mathcal{N}_s^p(s_i)|} \sum_{c_j \in \mathcal{N}_s^p(s_i)} \mathbf{W}_p^1 \mathbf{X}_{c[j]} \right) \quad (1)$$

$$\mathbf{h}_{c_j}^1 = \sigma \left(\mathbf{W}_{self}^1 \mathbf{X}_{c[j]} + \sum_{p \in \mathcal{P}} \frac{1}{|\mathcal{N}_c^p(c_j)|} \sum_{s_i \in \mathcal{N}_c^p(c_j)} \mathbf{W}_p^1 \mathbf{X}_{s[i]} \right) \quad (2)$$

where σ is an element-wise non-linear activation function (e.g., $\text{ReLU}(\cdot) = \max(0, \cdot)$ [13]), $\mathbf{X}_{s[i]}$ denotes the student features for s_i , $\mathbf{X}_{c[j]}$ denotes the course features for c_j , \mathbf{W}_{self}^1 is used to transform the self features from the original features, and \mathbf{W}_p^1 is used for transforming the features that are linked through the relation (i.e., course outcome type) p for the first layer.

–**Final Student Embeddings.** If we assume having L layers in our GCN model, we can then first define the last layer where we will obtain the student embedding $\mathbf{z}_i^s = \mathbf{h}_{s_i}^L$ for s_i as follows:

$$\mathbf{z}_i^s = \sigma \left(\mathbf{W}_{self}^L \mathbf{h}_{s_i}^{L-1} + \sum_{p \in \mathcal{P}} \frac{1}{|\mathcal{N}_s^p(s_i)|} \sum_{c_j \in \mathcal{N}_s^p(s_i)} \mathbf{W}_p^L \mathbf{h}_{c_j}^{L-1} \right)$$

where $\mathbf{h}_{s_i}^l$ represents the representation of student s_i at layer l of the GCN. Note that if we were to use a 2-layer GCN (i.e., $L = 2$) then $\mathbf{h}_{s_i}^{L-1} = \mathbf{h}_{s_i}^1$ would be coming from Eq. (1) and similarly $\mathbf{h}_{c_j}^{L-1} = \mathbf{h}_{c_j}^1$ from Eq. (2).

–**Final Course Embeddings.** If we assume having L layers in our GCN model, we can then first define the last layer where we will obtain the course embedding $\mathbf{z}_j^c = \mathbf{h}_{c_j}^L$ for c_j as follows:

$$\mathbf{z}_j^c = \sigma \left(\mathbf{W}_{self}^L \mathbf{h}_{c_j}^{L-1} + \sum_{p \in \mathcal{P}} \frac{1}{|\mathcal{N}_c^p(c_j)|} \sum_{s_i \in \mathcal{N}_c^p(c_j)} \mathbf{W}_p^L \mathbf{h}_{s_i}^{L-1} \right)$$

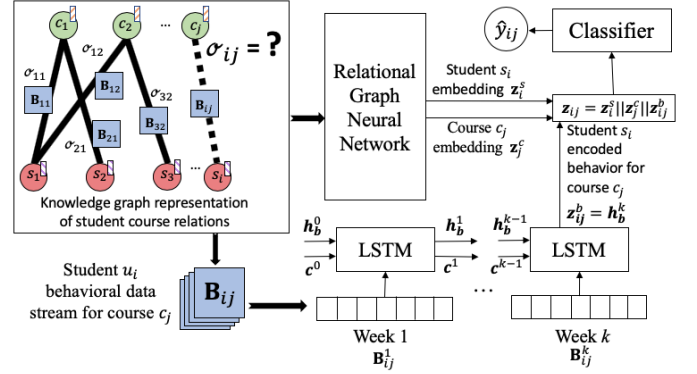


Figure 4: Visualizing the entire DOPE model consisting of both the relational graph neural network and recurrent neural network components.

where $\mathbf{h}_{c_j}^l$ is the embedding of student c_j at layer l of the GCN. Next, we will discuss how DOPE uses an RNN to encode a student's sequential behavior data associated with a given course.

3.3 Encoding Student Behavioral Data

In this part, we discuss how to encode the sequential edge features i.e., behavioural student data. To recall, when a student s_i is currently enrolled in a course c_j , by the k^{th} week we will have the features $\mathbf{B}_{ij} = [\mathbf{B}_{ij}^1, \mathbf{B}_{ij}^2, \dots, \mathbf{B}_{ij}^k]$. To better represent the behavioral data, we utilize a Long-Short Term Memory (LSTM) [16], which is an effective RNN variant that has been designed to extract temporal features from sequential data e.g., videos [25], speech [14, 25], and text [24, 23]. Furthermore, it has shown great abilities to capture temporal online user behaviors [26]. We fix the length of the behavior feature sequence for all students to be k (e.g., 10 weeks). Then for a given behavioral sequential data \mathbf{B}_{ij} , at each week $t \in [1, k]$, an LSTM unit takes the t -th week's click feature vector \mathbf{B}_{ij}^t as the input and uses LSTM formulation [16] to produce the output behavioral vector \mathbf{h}_b^t . The final output of the LSTM is $\mathbf{h}_b^k \in \mathbb{R}^b$ (i.e., output of last LSTM unit) when given the sequence \mathbf{B}_{ij} as input. Then, we set the encoded behavior of student s_i for course c_j as the e^b dimensional vector $\mathbf{z}_{ij}^b = \mathbf{h}_b^k$.

3.4 Final Course Performance Classifier

Here we combine student and course embeddings from the relational graph convolutional as well as encoded behavioral data and feed into a classifier. This can be seen in Figure 4. Given the student embedding \mathbf{z}_i^s for student s_i , course embedding \mathbf{z}_j^c for course c_j , encoded student behavior of s_i in the course c_j as \mathbf{z}_{ij}^b we form the final feature representation as follows:

$$\mathbf{z}_{ij} = \mathbf{z}_i^s || \mathbf{z}_j^c || \mathbf{z}_{ij}^b$$

where $||$ denotes concatenation and we concatenate the three components together into a single $(e^s + e^c + e^b)$ dimensional representation. For training DOPE, we use supervised learning such that labels are the outcome performances from the historical data $o_{ij} \in \mathcal{O}$ and matched with the training student and course pair (s_i, c_j) . More specifically, we construct a minibatch set \mathcal{M} that contains triplets of the form (s_i, c_j, T) where $T = o_{ij}$ (i.e., the course outcome type) and we assume the outcome type set \mathcal{T} where $|\mathcal{T}| = p$ since there are p course outcome types. The objective is then formalized in the following:

$$\mathcal{L} = \frac{1}{|\mathcal{M}|} \sum_{(s_i, c_j, T=o_{ij}) \in \mathcal{M}} \log \frac{\exp(\theta_T^{MLG} \mathbf{z}_{ij})}{\sum_{T' \in \mathcal{T}} \exp(\theta_{T'}^{MLG} \mathbf{z}_{ij})} + \lambda \text{Reg}(\theta^{RGCN}, \theta^{LSTM}, \theta^{MLG}) \quad (3)$$

where the classifier first maps \mathbf{z}_{ij} to a p dimensional vector through the parameters θ^{MLG} (since we have p different outcomes, i.e.,

Table 1: The description of the dataset.

Name	Train Periods	Test Periods	#Students
SS_1	2013J	2014J	735
SS_2	2013B, 2013J	2014B, 2014J	6622
SS_3	2013J, 2014B	2014J	2366
ST_1	2013B, 2013J	2014B, 2014J	5745
ST_2	2013J, 2014B	2014J	2685
ST_3	2013B, 2013J	2014B, 2014J	7092

link labels in the knowledge graph) and then utilizes the softmax function to get the outcome probabilities.

4. EXPERIMENTS

In this section, we conduct some experiments to verify the working of our proposed method.

4.1 Dataset and Experimental Settings

Online education platforms utilize virtual learning environments (VLEs) to collect records about all students' interactions and provide the opportunity for analyzing students' learning behavior. In this study, we use the data of The Open University Learning Analytics Dataset (OULAD) [29], which contains 22 open university courses for years 2013 and 2014 and 32,593 students. The dataset includes student demographic information, student assessment results, and daily interactions with the university's VLEs (10,655,280 entries). For each year, courses are offered in two distinct modules denoted as B and J (essentially they are similar to 'semester' in the conventional education system) where each module takes around 35 to 40 weeks long. The outcome of a course for a student can have four different categories including *Distinction*, *Pass*, *Fail*, and *Withdrawn*. We use OULAD and select three social science courses (i.e., SS_1 , SS_2 , and SS_3) and three Science, Technology, Engineering, and Mathematics (STEM) courses (i.e., ST_1 , ST_2 , and ST_3) as demonstrated in Table 1.

To represent the behavioral data, we count the different number of weekly clicks a student makes e.g., accessing resources, webpage click, forum click, quiz attempt, and so on. The size of each weekly behavioral vector is 20. Further, course attributes include two one-hot encoding vectors, one for representing a course among 6 courses, and the other one for holding either the course is social science or STEM. Train and test periods are shown in Table 1. We use 10% of the training data as a validation set to tune the hyper-parameters. The implementation is done using PyTorch package [30]. Each simulation is run for 200 epochs with a learning rate set to 0.001 and a decaying rate of 0.99 every 100 steps. As for the evaluation metric, we use weighted F1 score which is the harmonic mean of recall and precision.

4.2 Baseline Methods

We compare the performance of DOPE with the following baseline methods.

- **SVM.** In this baseline method, we concatenate the course attributes and students' demographic features as well as weekly click data (i.e., behavioral data) into a single vector and feed it to a support vector machine with radial basis function kernel.
- **LR.** This is similar to SVM except we use logistic regression for classification. The reason for including this baseline is to measure the online course performance prediction problem using a simple classification method without any kernel or non-linearity.
- **DOPE_{FCN}.** This is a variation of DOPE where instead of modeling behavioral data with an LSTM, we use a fully connected network. The reason for including this method is to evaluate the effectiveness of the way we model sequential behavioral data.

We compare DOPE with the baseline methods for the different numbers of weekly click data i.e., 5, 10, 15, and 20 weeks. By doing so, we can measure how effective DOPE is in the early prediction of a student's course performance prediction. We note that 20 weeks is almost half of a course period when there is still adequate time for intervention in the case of prediction as failure.

4.3 Binary Classification

As mentioned before, our dataset includes 4 distinct labels for a student's performance in a course, namely *Distinction*, *Pass*, *Fail*, and *Withdrawn*. In this section, we merge *Distinction* and *Pass* into a single class "Pass" and *Fail* and *Withdrawn* into a single class "Fail" and then perform a binary classification. Figure 5 illustrates the experimental results for all courses. We make the following observations based on the results presented in Figure 5.

- In general, the more weekly click data is introduced, the better we can predict the students' outcomes. DOPE enjoys more of such performance increase as compared to other methods. In particular, as early as 20 weeks from the start of a course (i.e., almost in the middle of a course duration), it can predict student's outcomes with very high performance. This allows teachers or online course administration to take actionable and interventive measures to help students with poor performance.
- DOPE achieves a better performance than DOPE_{FCN}. This shows the fact the LSTM component as a machinery extracting temporal features from click behaviors is necessary and affects the model's predictive power.
- DOPE is shown to be effective for all courses as we can observe it achieves an F1 score of more than 0.8 across all courses when 20 weeks of click data are considered.

4.4 4-class Classification

In this part, we compare the performance of DOPE with baseline methods for a 4-class classification setting whose experimental results are demonstrated in Figure 6. We make the following observations based on the results in Figure 6.

- The observations we made for binary classification hold for 4-class classifications as well. In particular, DOPE still outperforms baseline approaches, more weekly click data is helpful in course outcome prediction, and the LSTM can effectively handle sequential that than simple concatenation followed by a fully connected network model (i.e., DOPE_{FCN}).
- Since more classes are considered, compared to binary classification, the 4-class classification is a harder task. In particular, now *Withdrawn* is considered as a separate class, which might be "conceptually" hard for a model to discern from *Fail*.

4.5 Behavioral Feature Analysis

Since behavioral data (i.e., click data) plays an essential role in determining a student's performance, we conduct a feature analysis experiment investigating the importance of each behavior type. A similar feature analysis has been performed to discover great insights into human behaviors [21]. To this end, we follow an ablation feature analysis where at each time we include one feature type and suppress the rest (setting their values to zero) and then acquire the F1 score from the model. We do this experiment for the binary classification and the case when 20 weeks of click data is included. Figure 7 demonstrates the results and we make the following observations accordingly.

- For all courses, feature type *homepage* is associated with a high F1 score. This seems reasonable since most of the click activity occurs on the main page of the platform interface.
- Interestingly, clicks and activities in *forums* have an influential role in predicting fail or pass of a student in a course. This is in line with previous [15, 31] where they showed that MOOC forum activities correlate with a student's academic performance.

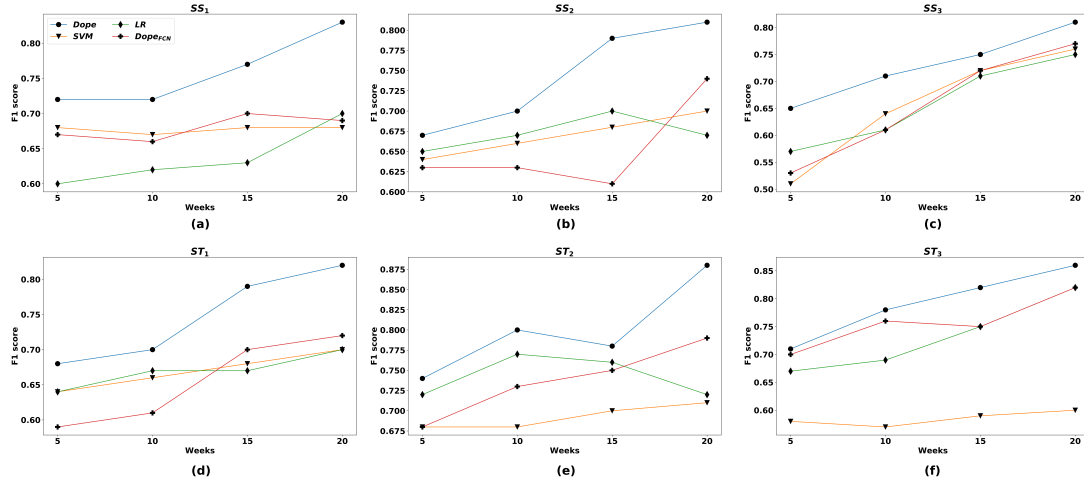


Figure 5: Comparison results for binary classification using four different amounts of included weekly click data.

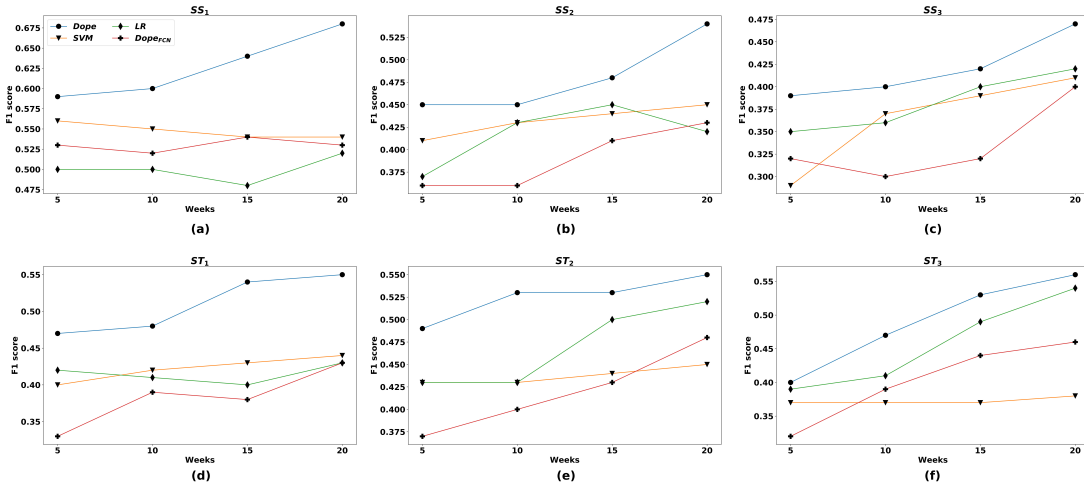


Figure 6: Comparison results for 4-class classification using four different amounts of included weekly click data.

- Unique behavioral importance profile of a course can aid policymakers, administrators and even course interface designers to prepare the course materials in a more informed way. For instance, we can observe that attribute *wiki* is playing an important role in performance prediction of ST_2 while its effect is negligible for other courses. This can be indicative of materials of the course ST_2 to be requiring more wiki access and consequently, the content can be changed accordingly.

Based on the observations above, we can conclude that DOPE encodes behavioral data in an intuitive manner that conforms to previous studies' findings as well.

4.6 Inter-course Outcome Evaluations

Naturally, each course has its own model. However, in this section, we intend to measure inter-course performance evaluation where we train DOPE on one course and test it on another one. Table 2 shows the results. Again, the models are trained for the binary classification and they incorporate 20 weeks of the click data. Also, for the reference, we have included intra-course performance (i.e., the same course for training and test) shown in the diagonal entries of Table 2. Expectedly, when the training course and the test course are the same (i.e., intra-course setting), the

performance is higher. This seems reasonable since clicking patterns are expected for the course in the past (i.e., a part of the training data) and the one in the future (i.e., testing data), and the model can more easily extract such patterns. Although the results for inter-course results are not as good as the ones for intra-course, we still see that the DOPE can effectively achieve reasonable performance. This indicates that the proposed model DOPE can detect salient click and demographic patterns that are transferable from a course to another.

Table 2: Inter-course performance evaluation

	Test course					
	SS_1	SS_2	SS_3	ST_1	ST_2	ST_3
SS_1	0.83	0.78	0.77	0.71	0.8	0.75
SS_2	0.63	0.80	0.58	0.66	0.53	0.66
SS_3	0.64	0.51	0.80	0.45	0.72	0.49
ST_1	0.60	0.79	0.71	0.82	0.47	0.41
ST_2	0.74	0.60	0.56	0.62	0.88	0.49
ST_3	0.79	0.76	0.75	0.70	0.77	0.86

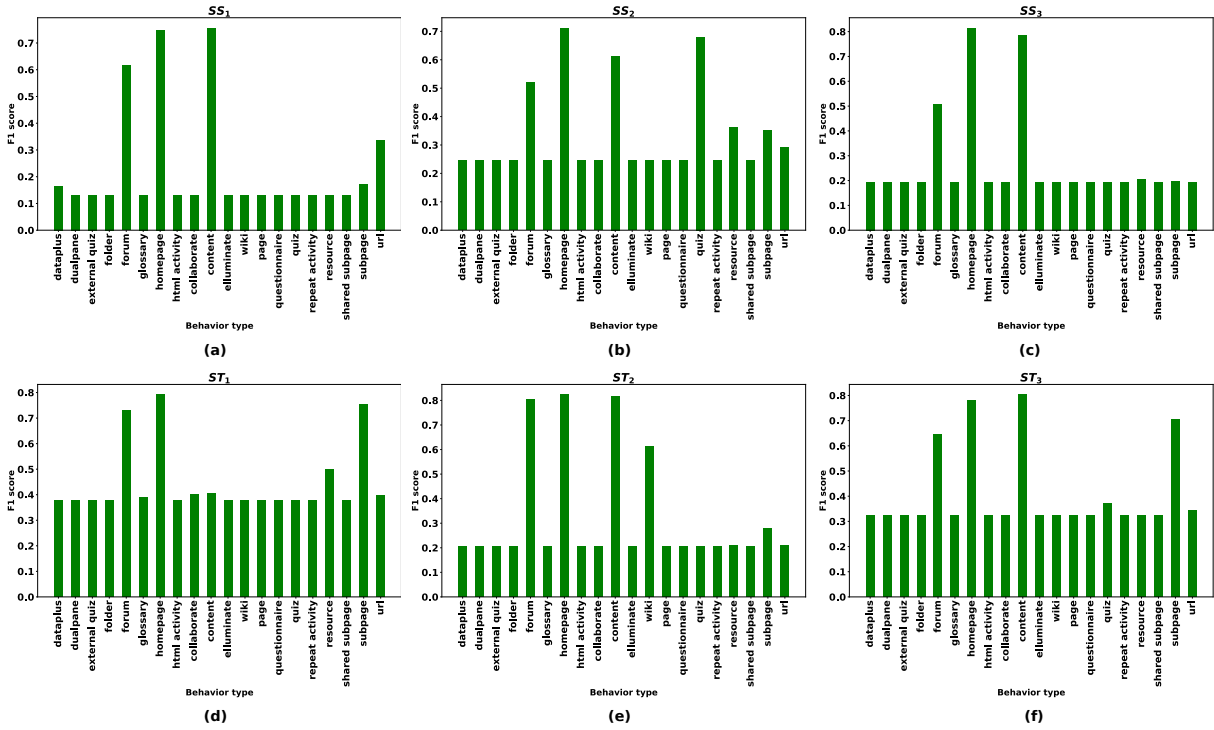


Figure 7: Behavioral feature analysis on different courses. The models are binary classification using 20 weeks of click data.

5. RELATED WORK

In the following, we highlight some of the works focusing on student dropout and performance prediction. In [35], they extracted 27 interpretive features and used logistic regression to predict student persistence prediction. The authors of [32] used probabilistic soft logic to model student survival by constructing probabilistic soft logic rules and associating them. Different from [32] (which mainly considered forum features), in [28] they did not consider forum data, but instead only made use of clickstream data to train their prediction model. More specifically, they used principal component analysis [37] paired with a linear support vector machine [4] for each week. It was in [12] that a more comprehensive approach was taken that used standard classification trees [2] and adaptive boosted trees [1] to construct their two-stage Friedman and Nemenyi procedure for dropout prediction by processing different features such as clickstream-based, forum-based, and assignment-based features. More recently, in [3], the authors studied a hybrid method for dropout prediction by combining both a decision tree [11] and extreme learning machine [18]. In addition to these traditional machine learning methods, some researchers have tried to use different deep learning models for dropout prediction of online courses. In [9] an LSTM was used to deal with the features extracted from students' interaction with lecture videos, forums, quizzes, and problems. [36] explored the potential benefits of employing a fully connected feed-forward neural network for dropout prediction. Different from previous work, [10] proposed a context-aware feature interaction network to incorporate context information of both participants and courses. More specifically, they used an attention-based mechanism for learning activity features. The most similar method to ours is found in [17] where they sought to conduct performance evaluations on students using a graph neural network (GNN), but there are primary differences: (1) they constructed separate small graphs of courses for each student while DOPE constructs a single knowledge graph of historical student course relations; (2) their graph neural network was used to obtain a graph classification for a given student based on that student's specific course graph, while our method uses the relational graph neural network to learn embeddings for both students and courses

from a single large knowledge graph; and (3) DOPE furthermore utilizes an LSTM model to capture a student's rich sequential behavioral data beyond just using static fixed student features.

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a model for course performance prediction we call it Deep Online Performance Evaluation (DOPE). Our method first represents the online learning system as a knowledge graph, such that we then learn student and course embeddings from historical data using a relational graph neural network. Simultaneously, DOPE utilizes an LSTM for harnessing the student behavior data into a condensed encoding, as the data has a natural inherent sequential form. We tested the proposed model on six courses from the OULAD dataset where the results showed the feasibility of DOPE and that it can predict at-risk students of on-going courses. We also investigated the usefulness of the different types of behavioral features and observed that DOPE encodes the data in an intuitive manner.

In the future, we will first analyze the imbalance and sparse issues of the dataset. One possible way to alleviate the sparsity would be through a network alignment [6] of multiple MOOC datasets represented as knowledge graphs or connecting student behavior data from social media for better predictions in online education [22]. Also, we will investigate more advanced ways of handling behavioral data. For example, investigating better ways to use "subpage" clicks beyond a simple aggregation that ignores separating the multiple different "subpages". In addition, we plan to apply our framework to the traditional education system aiming at identifying similarities and differences between online and traditional course performance prediction, since we believe this to be highly important in improving online learning systems.

Acknowledgement

This research is supported by the National Science Foundation (NSF) under grant numbers IIS1907704, IIS1928278, IIS1714741, IIS1715940, IIS1845081, and CNS1815636.

7. REFERENCES

- [1] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.
- [2] W. Buntine. Learning classification trees. *Statistics and computing*, 2(2):63–73, 1992.
- [3] J. Chen, J. Feng, X. Sun, N. Wu, Z. Yang, and S. Chen. Mooc dropout prediction using a hybrid algorithm based on decision tree and extreme learning machine. *Mathematical Problems in Engineering*, 2019, 2019.
- [4] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [5] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.
- [6] T. Derr, H. Karimi, X. Liu, J. Xu, and J. Tang. Deep adversarial network alignment. *arXiv preprint arXiv:1902.10307*, 2019.
- [7] T. Derr, Y. Ma, W. Fan, X. Liu, C. Aggarwal, and J. Tang. Epidemic graph convolutional network. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, page 160–168. ACM, 2020.
- [8] T. Derr, Y. Ma, and J. Tang. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 929–934. IEEE, 2018.
- [9] M. Fei and D.-Y. Yeung. Temporal models for predicting student dropout in massive open online courses. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 256–263. IEEE, 2015.
- [10] W. Feng, J. Tang, and T. X. Liu. Understanding dropouts in moocs. *Association for the Advancement of Artificial Intelligence*, 2019.
- [11] J. Friedman, T. Hastie, R. Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.
- [12] J. Gardner and C. Brooks. Dropout model evaluation in moocs. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [13] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.
- [14] A. Graves, N. Jaitly, and A.-r. Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE, 2013.
- [15] C. He, P. Ma, L. Zhou, and J. Wu. Is participating in mooc forums important for students? a data-driven study from the perspective of the supernet. *Journal of Data and Information Science*, 3(2):62–77, 2018.
- [16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [17] Q. Hu and H. Rangwala. Academic performance estimation with attention-based graph convolutional networks. In *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, volume 168, pages 69–78. ERIC, 2019.
- [18] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, volume 2, pages 985–990. IEEE, 2004.
- [19] N. Jha, I. Ghergulescu, and A.-N. Moldovan. Oulad mooc dropout and result prediction using ensemble, deep learning and regression techniques. 2019.
- [20] K. Jordan. Massive open online course completion rates revisited: Assessment, length and attrition. *International Review of Research in Open and Distance Learning*, 16(3), 2015.
- [21] H. Karimi*, T. Derr*, A. Brookhouse, and J. Tang. Multi-factor congressional vote prediction. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 266–273, 2019.
- [22] H. Karimi, T. Derr, K. Torphy, K. Frank, and J. Tang. A roadmap for incorporating online social media in educational research. *Teachers College Record Year Book*, (2019), 2019.
- [23] H. Karimi, P. Roy, S. Saba-Sadiya, and J. Tang. Multi-source multi-class fake news detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1546–1557, 2018.
- [24] H. Karimi and J. Tang. Learning hierarchical discourse-level structure for fake news detection. In *Proceedings of the 2019 Conference of the NAACL-HLT, Volume 1*, pages 3432–3442, 2019.
- [25] H. Karimi, J. Tang, and Y. Li. Toward end-to-end deception detection in videos. In *2018 IEEE International Conference on Big Data*, pages 1278–1283, 2018.
- [26] H. Karimi, C. VanDam, L. Ye, and J. Tang. End-to-end compromised account detection. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 314–321, 2018.
- [27] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [28] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart. Predicting mooc dropout over weeks using machine learning methods. In *Proceedings of the EMNLP 2014 workshop on analysis of large scale social interaction in MOOCs*, pages 60–65, 2014.
- [29] J. Kuzilek, M. Hlosta, and Z. Zdrahal. Open university learning analytics dataset. *Scientific data*, 4:170171, 2017.
- [30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.
- [31] B. K. Pursel, L. Zhang, K. W. Jablowski, G. Choi, and D. Velegol. Understanding mooc students: motivations and behaviours indicative of mooc completion. *Journal of Computer Assisted Learning*, 32(3):202–217, 2016.
- [32] A. Ramesh, D. Goldwasser, B. Huang, H. Daume III, and L. Getoor. Learning latent engagement patterns of students in online courses. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.
- [33] J. L. Santos, J. Klerkx, E. Duval, D. Gago, and L. Rodríguez. Success, activity and drop-outs in moocs an exploratory study on the uned coma courses. In *Proceedings of the Fourth International Conference on Learning Analytics and Knowledge*, pages 98–102. ACM, 2014.
- [34] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.
- [35] C. Taylor, K. Veeramachaneni, and U.-M. O’Reilly. Likely to stop? predicting dropout in massive open online courses. *arXiv preprint arXiv:1408.3382*, 2014.
- [36] J. Whitehill, K. Mohan, D. Seaton, Y. Rosen, and D. Tingley. Delving deeper into mooc student dropout prediction. *arXiv preprint arXiv:1702.06404*, 2017.
- [37] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.
- [38] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.
- [39] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.

EDM and Privacy: Ethics and Legalities of Data Collection, Usage, and Storage

Mark Klose
North Carolina State University
Raleigh, NC, USA
mwklose@ncsu.edu

Vasvi Desai
North Carolina State University
Raleigh, NC, USA
vcdesai@ncsu.edu

Yang Song
UNC at Wilmington
Wilmington, NC, USA
songy@uncw.edu

Edward Gehringer
North Carolina State University
Raleigh, NC, USA
efg@ncsu.edu

ABSTRACT

Imagine a student using an intelligent tutoring system. A researcher records the correctness and time of each of your attempts at solving a math problem, nothing more. With no names, no birth dates, no connections to the school, you would think it impossible to track the answers back to the class. Yet, class sections have been identified with no more data than this. This paper recounts shocking episodes where educational data was used to re-identify individual students, build profiles on students, and commit fraud. We look at the ethical principles that underlie privacy as it relates to research data, and discuss ethical issues in data mining relating to social networks and big data. We explore four major types of data used in EDM: (i) clickstream data, (ii) student-interaction data, (iii) evaluative data, and (iv) demographic data. Each type of data can be harmful if disclosed in particular contexts, even if all personally identifiable information is removed. We consider laws and legal precedents controlling access to student data in the United States and the European Union. This paper concludes by describing some practical situations in EDM and suggesting privacy policies that satisfy the ethical concerns raised earlier in the paper.

Keywords

Privacy, anonymization, de-identification, ethics, educational data mining

1. OUR DATA ARE MORE THAN VALUABLE

Educational data mining (EDM) analyzes student data from Learning Management Systems (LMSs) and stand-alone educational applications. Educational technology (EdTech) vendors use student data to analyze student performance, improve student models, and discover opportunities to boost

learning. Any EdTech data breach or unjustified student tracking infringes student privacy, generates huge controversy, and produces big headlines. The ability to create auxiliary connections with other known information makes data valuable to both hackers and researchers. EDM researchers need to understand privacy risks raised by sensitive data.

1.1 Privacy risks of educational data breaches

One of the biggest leaks of student data was the Edmodo data breach. Edmodo is an EdTech company that provides coaching tools and a collaborative platform for K-12 students and teachers to communicate about course content, quizzes and assignments. The breach involved 11.7 gigabytes of data and over 77 million uniquely identifiable users, exposing at least 50 million usernames and 29 million emails. Edmodo did acknowledge the breach's occurrence, but by that time data was being sold by the hackers on the black market [9]. The breach was important, not because of the inherent value of the data itself, but rather because of how the data could be connected with auxiliary datasets. Having a list of hashed passwords is not useful; knowing that people tend to reuse passwords exposes other systems to greater risk. Leaking names and email addresses also left the students at greater risk of identification or additional tracking.

Since other breaches or publicly available datasets reveal personal information such as addresses or ethnicity, they can be cross-referenced to the leaked data to reveal a more complete identity. Companies shy away from controversy when it places their product or service at risk, and victims tend to not come forward, lest they sacrifice their privacy. It is important to take these breaches seriously, as any data leaked by research projects or products has more value now than before this large data breach.

Let's examine other educational data breaches. At Torrey Pines High School in San Diego, California, the online grading system was hacked to alter students' grades and transcripts [17]. This incident highlights risks like grade changes by unauthorized parties. In Montgomery County, Maryland, a student performed a brute-force attack on Naviance, an online platform for college and career readiness. The attack exposed sensitive data from 5962 accounts, including names,

Mark Klose, Vasvi Desai, Yang Song and Edward Gehringer "EDM and Privacy: Ethics and Legalities of Data Collection, Usage, and Storage" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 451 - 459

addresses, phone numbers, GPA, and SAT scores [12], that students trust will remain private and protected.

In the last two years, Chinese media have reported several cases where students' personal information, including their national identity card number, was stolen or leaked. Companies then used students' identities as phantom employees for tax fraud [7, 8]. One of the saddest cases related to an educational data breach occurred in 2016. After the national college entrance exam, a criminal group hacked a local university application system and acquired students' personal information, including phone numbers. The perpetrators posed as financial-aid officers then contacted students, asking them to transfer money into specified accounts before their financial aid could be delivered. One student contacted, Yuyu Xu, died from sudden cardiac arrest after discovering that it was a fraud [6].

1.2 De-identification is not enough

In a 2005 contest to devise better movie-recommendation systems, Netflix released 10 million movie rankings by 500,000 customers after removing all direct customer-related information. Two University of Texas researchers showed that simple anonymization fails to preserve privacy; researchers connected anonymized Netflix dataset entries with distinct users in the Internal Movie Database [43]. Similar risks are present in anonymized datasets used for EDM research.

Another compromise of anonymized data occurred when the "Tastes, Ties, and Time" (T3) project released de-identified Facebook profile data. All personally identifiable data was removed, such as names, email addresses, university name, and names of friends. However, the dataset's associated code book provided a list of students' majors and state or country of origin. Within a couple of days, researchers at University of North Carolina at Chapel Hill and University of Wisconsin-Milwaukee identified the "anonymous northeastern university" as Harvard. This raised significant privacy concerns as research assistants at Harvard University who were "friends" with some students in question, had deeper access to profiles than the general public. Both the Harvard IRB and Facebook had approved the project [69].

In one case, researchers collected clickstream data from multiple classrooms across the country instead of personally identifiable information (PII) or demographic data about an individual student. They created clusters of students from log files, recording time and correctness of students' responses. These clusters were enough to identify classes of gifted students and extract demographic data about student groups [67]. When one cluster missed a day's worth of work, the researchers cross-referenced potential classrooms with announcements of a field trip. This resulted in a single classroom being identified using only anonymized data.

Common de-identification techniques include: *Anonymization*, where all PII is simply removed from the dataset; *hashing*, where multiple fields (e.g., last name and email address) are hashed into a single value, and replace the original fields in the record; *swapping*, where some field, such as a name, is switched to apply to someone else's record; and *noising*, where data values are perturbed (changed) in some way [37].

These common de-identification techniques can have adverse effects on data quality [14]. Protecting students' privacy by removing re-identifiable attributes from data can reduce the data's utility for analysis [67]. Noising data can diminish performances of supervised learning models [44]. Despite not changing aggregations, swapping has similar effects [41]. Thus, it is crucial to balance privacy with utility. Ohm (2009) warns that "the utility and privacy of data are linked, and so long as data is useful, even in the slightest, then it is also potentially re-identifiable" [47].

Building profiles for targeted advertising also endangers student data privacy. Data can be collected by amassing emails or system interactions, like websites visited. Students can be identified and targeted on the basis of their answering patterns, e.g., what questions they answered correctly. Google, which provides the educational content platform GSuite for Education, has been alleged to have built personalized profiles of students based on their GSuite interactions, and to have scanned students' emails to target advertising [23, 28]. Selling the data to third-party vendors without consent would raise severe ethical questions.

In these cases, students' identities and information were used or revealed without explicit permission, undermining the idea of consent. Although releasing someone's homework grade or test score seems trivial, researchers need clear understanding of what constitutes legal and ethical usage of student data so students remain protected while the EDM research efforts continue into noble frontiers.

2. ETHICAL PRINCIPLES RELATING TO EDUCATIONAL RESEARCH DATA

Deciding what constitutes "responsible use" bridges research ethics and other ethics subfields. Each field emphasizes different aspects of the research process. As with any ethical discussion, clarifications of these terms and new realizations of technologies causes these principles to evolve to reflect the state of EDM research. Several analyses have looked at key principles in a more abstract form [48, 51, 59], but these works are too broad to answer specific questions. This paper seeks to highlight key principles from each subfield and applications to specific areas of EDM research.

2.1 Research ethics

Much of the literature on research ethics derives from the Nuremberg Code [34], the Helsinki Declaration [2], and the Belmont Report [15, 35, 65]. But the Belmont report lacks specifics on internet-mediated research [1]. The Menlo Report [4] extends principles from the Belmont Report to computing centric research. It adopts three principles found within the original Belmont Report, and adds a new fourth principle, respect for law and public interest.

2.1.1 Respect for Persons

The Belmont Report establishes the principle of *respect for persons* through two key frames: treating individuals as autonomous agents and entitling individuals to protections [65]. The Menlo Report adds consideration of computer systems and data that directly impact people who are typically not research subjects themselves [4]. This impacts the concept of informed consent. Informed consent comprises three

concepts: notice, comprehension and voluntariness [4]. For EDM research, consent documents must not promise improved service or instruction in return for participation; this could be interpreted as coercion. This is relevant to intelligent tutoring systems (ITSs)—students unwilling to allow an ITS to use their data for research purposes should not thereby be academically disadvantaged.

The Menlo Report reiterates consent from one person does not constitute consent from all members of their group, and consent given for one research purpose should not be considered valid for different purposes. Since data subjects are co-owners of educational data [40], concepts such as downstream consent [13] should be considered for applications like educational data warehouses. To further protect individuals, the Menlo Report suggests de-identification of data. De-identified data can fit into the special regulatory category of “pre-existing public data,” which affords more opportunity for exemptions granted by research ethics boards.

2.1.2 Beneficence

For identifying of potential benefits and harms, the Menlo Report targets systems assurance (confidentiality, availability, integrity) and individual and organizational privacy [4]. Within EDM, this means identifying likely flaws or biases in ITSs prior to deployment or introducing protections for model inference and model inversion attacks [20, 52, 56, 58].

When collection or storage of high-risk data is necessary, the Menlo Report suggests to destroy data once past the retention period of scientific reproducibility, which is commonly 3 years at minimum [11, 46]. A tension exists between data retention for research replication and ensuring privacy of data subjects, which will be discussed further in section 3.2. Utilizing data aggregations prevents the need to store sensitive information that could tie back to a specific student or class.

2.1.3 Justice

With regard to *justice*, the Menlo Report declares research should not target specific people or groups based on attributes such as technical competency or personal demographics [4]. For EDM researchers creating some model or product, this discourages using convenience samples such as classrooms the researcher worked with previously. Instead, research should target classrooms or groups of students where the potential intervention provides the most benefit. Using prior data providing an accurate cross-section of the larger community being studied is more favorable than potentially excluding future groups from participation.

The Menlo Report compares actively excluding groups out of prejudice and actively including entities willing to cooperate and consent. Including entities demonstrates the principles of Respect for Persons and Beneficence outlined earlier. Specifically targeting subjects through coercion undermines legitimate research and violates the principle of Justice [4].

2.1.4 Respect for Law and Public Interest

The Belmont Report implicitly classifies respect for the law and greater public interest as an aspect of Beneficence. The Menlo Report considers it a fourth principle with two applications: compliance and transparency/accountability [4].

These provide some assurance of public good whenever identifying stakeholders is difficult or impossible. Lacking transparency and accountability weakens current research projects at hand and learning analytics research credibility as a whole.

Within EDM research, compliance, transparency, and accountability all require researchers to understand relevant laws in their jurisdictions. Researchers are culpable for being up-to-date on laws and regulations where they perform research. Transparency means releasing source code or clearly communicating what information is collected and what computations are performed. Transparency is in the interest of research subjects, the beneficiaries of research, and research ethics boards as they audit projects where necessary.

2.2 Social networks and ethics

With a growing level of research incorporating data directly from social networks, EDM must consider the various ethical principles guiding online behaviors. The disconnect of individuals from online identities must be considered while using social networking data and its derivations. Seeing incomplete aspects of an individual’s personal life through their social network lens affects and alters the perception of them.

Users curate their identities in an online setting [61]. Some students may only use social network services to communicate within specific spheres like family, workplaces, or friends. Students’ online actions and behaviors may not truly reflect themselves as learners, but as reflections of the sphere they are in. Social networks have distinct group dynamics, similar to the real world, further complicating the trustworthiness profiles provide as a snapshot of the student. In theory, social network users should be exposed to opinions of diverse worldwide users, but in practice, views and news feed algorithms constrict types of content users see [50]. In online settings, users tend to subjugate their identities to the group identity they participate in (e.g., student, liberal, conservative, Christian, Muslim), in order to conform to the group [50]. With these considerations in mind, this may devalue the student’s social network presence to the point where social network data may lack enough integrity to be used.

Some broad concerns with using social network data include availability of users’ data to third parties to create marketing profiles, using data mining applications without their knowledge or consent, surveillance by law enforcement, or having third party applications collect and publish user data without notification [66]. Social networking services provide privacy controls for users; however, failure to understand implications of sharing information on a social networking service results in decreased privacy for users in relation to outside actors such as researchers [5].

When releasing de-identified Facebook account data as part of the T3 project, researchers placed limited concern on research ethics and students’ privacy. Utilizing data was not the problem; failing to recognize how collection methods affected privacy is the issue. While acquiring profile data, researchers could have broader access than originally intended by the profile owner. This happens if researchers have prior connections through memberships in their organization or having mutual connections to the profiles. If research combines educational and social networking data but disrespects

privacy standards laid out by the student on platforms like Facebook (or if researchers fail to seek further consent from students about using their social network data), then this breaches the student's overall privacy [69].

2.3 Big-data ethics

Data possesses properties distinguishing itself from other advanced forms of technology, not limited to: its regarding as an aspect of societal infrastructure; its interconnectedness; its dynamic nature for discovery beyond original purpose; its real-time analysis and decision-making possibilities; its usability regardless of where, when, and for what purpose it was collected; its reusability for unexpected purposes to reveal unexpected information (the core purpose of data mining); its intrusiveness due to storing data about individuals in multiple databases; its ownership issues, especially in education settings [26, 40]. Each individual datum is useless without context and associated metadata. By this construction, value added to data provides its potential for misuse. In EDM, positive outcomes for students come from discrete products or further insight on learning motivations and processes; this does not exclude potential misuse by researchers.

Although this paper will not discuss algorithmic fairness in detail, the concept applies in big data ethics. Many practices classify or regress individual experiences into common baselines based on socioeconomic status, race, ethnicity, or gender without explanatory data. Division into classification groups and averaging metrics stereotypes students. Unchecked stereotyping could rehash old prejudices that negatively affect research itself. The lack of care to blindly use basic classification groups can be extreme enough to break the principle of “doing good work” [26]. Instead, research should favor groups created by methods like Topological Data Analysis [22] and other Bayesian models that cluster outside of traditional demographic groupings.

2.4 Ethical uses of specific educational data

There are four kinds of data commonly used in EDM that have further ethical concerns for researchers: clickstream, student interaction, evaluative, and demographic data.

2.4.1 Clickstream data

At minimum, clickstream data provides information that some generic user initiated an interaction at a given time. Although relatively safe on its own, a classroom worth of students generating clicks can reveal the location of the classroom, especially if the classroom functions on a daily or weekly routine. Studies have already shown tracking IP addresses to reveal geolocation [38], which shows the potential for this to be done for clickstream data as well. Utilizing this aspect of clickstream data allowed Yacobson et al. to identify a gifted student classroom from a completely anonymized dataset of over 500 students after clustering time of clicks and correctness of answers. Once a class deviated from the schedule due to a field trip, researchers then identified the school and classroom in question [67].

2.4.2 Student interaction data

Student interaction data includes peer assessments, online discussion forums, and team-member evaluations—data with a clear writer-respondent relationship. These interactions

directly reflect the respondent's viewpoints, which may violate privacy if shared outside of the student-teacher relationship when they cast aspersions on the student. If negative comments are given about the writer, and that information somehow leaves the model or is revealed to an outside source, this could affect student's relationships and future prospects. Similarly, sharing class forums regarding sensitive subjects like sexuality to wider audiences is not proper since this could potentially identify and harm a student.

2.4.3 Evaluative data

Evaluative data references include grades and other inputs to predictive analytics models. In educational settings, clear benefits to predictive models include quality assurance and improvement of instruction, tracking and predicting retention rates, and enabling the development of adaptive learning [3, 57]. These same models could influence later interactions between students and instructors, thus affecting the relationship and trust—a proven factor in the academic success of students [19, 21, 39, 53]. For example, if a predictive model flags a student for high potential of failure and dropout from a course, the instructor may focus interventions on that student. This could overcome other reasons for a student's poorer performance, thus remedying symptoms rather than determining underlying causes for the struggles.

2.4.4 Demographic data

Many studies looked at how simple demographics can identify a non-negligible number of individuals [24, 60]. Student demographic data can be combined with other data to infer identities of students. In the T3 project, student Facebook data identified many individuals as being the only Harvard freshman student from a certain state or country. Identifying an exact student is possible when combining news announcements or other university materials [49]. With the growing number of data breaches like the Edmodo case and the noted intrusiveness of big data due to an individual's membership in many databases, this leads to a risk that often goes unnoticed for smaller research applications. For EDM researchers, having researchers redact some demographic information, when not integral to research, may assist students in controlling their information and privacy.

In summary, it is vital to see how ethics does not adhere to data itself; the researchers themselves and how the data are used carries significant ethical implications. Understanding the scope of data collection, storage, and usage ultimately impacts the ethics of research and benefits for learners.

3. LAWS AND LEGAL PRECEDENTS

Legal regimes vary substantially throughout the world; an exhaustive comparison is beyond the scope of this paper. Legal frameworks for educational data exist in other countries, but their impacts are less clear [18, 62, 63, 64]. We focus on the two largest EDM research communities: the United States and the European Union.

3.1 United States

In the United States, the most relevant legislation is the Family Educational Rights and Privacy Act (FERPA). Enacted in 1974 after widespread concern about intrusive psychological testing of students, FERPA defined the circum-

stances which allow schools to release a student’s “education records” to outsiders (including EDM researchers). Personally identifiable information (PII) about a student must not be disclosed unless the student, or the parents if the student is under 18 years old, give their prior consent. The law applies to all schools that receive funds under a program administered by the US Department of Education. In practice, this includes virtually all colleges and universities, as well as public (but not private) elementary and secondary schools.

Under the law, PII includes students’ names, names of parents and family members, Social Security or student ID numbers, biometric records, and other indirect identifiers including date or place of birth and mother’s maiden name. It also includes “[o]ther information that, alone or in combination, is linked or linkable to a specific student that would allow a reasonable person in the school community, who does not have personal knowledge of the relevant circumstances, to identify the student with reasonable certainty.”

This raises two main issues: What is an education record, and what does the “other information” mentioned above include? For example, does clickstream data collected by an ITS count as an “education record”? The most relevant court case is *Owasso v. Falvo* (534 US 426 (2002)). This case arose in Oklahoma, when a teacher had students peer-grade each other’s papers. Papers were collected from students and passed out to other students. The teacher called out the correct answers, and each student would mark answers on the paper in front of them as correct or incorrect. The school district was sued by the a student’s mother who said that her son, who had not scored very well, had been embarrassed when a fellow student called out his score.

The case eventually reached the US Supreme Court, which ruled unanimously that peer grades did not constitute “educational records.” FERPA established a two-part test to determine what was an educational record: (i) The material must “directly relate to the student” and (ii) must be maintained by the institution or an individual acting on the institution’s behalf. The decision turned on the test’s second part. The court ruled grades were not “maintained by .. an individual acting on behalf of the institution,” at least until entered in the teacher’s gradebook. The court did not rule whether teachers’ gradebooks are an educational record.

The *Owasso* decision seems to imply that FERPA does not prevent the disclosure of student classwork and homework to outside researchers, except possibly if outsiders can use it to discover students’ grades for the assignment. In general, data from web-based participatory learning tools is not covered under FERPA [27]. Note that this is a legal judgment, not an ethical one, since disclosure of student information from some such tools may allow re-identification by others.

However, one clause in FERPA implies this situation may not last. The clause on linkable information implies that what constitutes PII changes as technology changes [68]. As datasets become higher dimensional, the possibility of using an auxiliary dataset to re-identify people grows [42]. Thus, every researcher releasing a de-identified dataset should be familiar with the growing risks.

A distinction should also be made between datasets used for analytics and datasets used for intervention [30]. A researcher simply analyzing effects of some practice or tool on student learning has little need to track individual identities. If the dataset is used for intervention—to improve experiences of particular students—obviously the students’ identities must be preserved. In this case, FERPA may still apply, since neither the law nor *Owasso v. Falvo* clearly delineates what kind of research data constitutes an “educational record.” Fortunately, interventions are often in house; data of this nature would rarely be important to outside researchers. However, if interventions are with students in other institutions, it would be worth seeking legal guidance.

3.2 European Union

The European Union adopted the General Data Protection Regulation (GDPR) in April 2016, which took effect in May 2018. GDPR applies to processing “personal data” tied to an identifiable person. For practical purposes, this seemingly is the same as FERPA [45] (except GDPR also applies outside educational contexts). According to GDPR [33],

[A]n identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an on-line identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person.

This is subject to the same uncertainties as FERPA. One place where the two laws differ is in the EU, the subject must consent to use of their personal data: the researcher must secure “a clear affirmative act establishing a freely given, specific, informed and unambiguous indication of the data subject’s agreement to the processing of personal data relating to him or her ... Silence, pre-ticked boxes or inactivity should not ... constitute consent [recital 32]” [29].

Another consideration is the GDPR’s signature provision: the “right to be forgotten.” If the subject of the data withdraws consent, the data must be erased. The data must also be erased when no longer needed for the purpose (e.g., research) that it was collected for [29].

4. IMPLICATIONS OF EDM RESEARCH

Having discussed privacy risks, ethical considerations, and legal risks for EDM researchers, we now examine current privacy concerns and work needed in the near future. Through correspondence with EDM 2019 researchers and reflection on our own research, we identify the following areas as requiring attention to the principles and risks established earlier.

4.1 Crawling learners’ data outside a platform

Though most EDM researchers use data generated *within* educational platforms/systems such as MOOCs/ITSs, sometimes it can be tempting to acquire data on learners *beyond* a specific tool and beyond the course duration. When researchers use the learners’ information to access their data on social web platforms after they have finished a MOOC,

for example, more research questions can be answered, such as “does displaying MOOC certificates have an impact on learners’ career paths?” Chen et al. traced the learners’ profile data overtime on Gravatar, StackExchange, GitHub, Twitter, and LinkedIn after the MOOCs to investigate the impact of MOOCs in the long-term [10]. Chen et al. used data from 18 MOOCs and reported they could reliably identify the highest of 42% of the learners in a MOOC on social web platforms. The MOOC data (from edX) they started with had the usernames, full names, email addresses. Therefore, it is not a surprise that a high percentage of learners can be identified. However, crawling data of learners on five social-media platforms several years after they have finished a MOOC does bring up privacy concerns.

Arguably, learners’ profile and posts are the data available to the public, but EDM researchers are able to join learners’ data from an educational platform with learners’ data on social web platforms, which may give researchers too much power in mining learners’ data after they have finished their learning. Considering the learners’ additional data can potentially be crawled, the sharing and reusing learners’ data should be backed with appropriate legal agreements. For example, Yacobson et al. suggested to “ban linking application data with external data sources” [67].

4.2 Community consensus on learners’ privacy

Researchers need data with high utility, but the effort to anonymize data hurts this. In other words, keeping datasets of high utility and high privacy level concurrently is hard. De-identification protects learners’ privacy, but too strict de-identification can negatively affect analysis [37].

Besides, there is always a risk that de-identified data can be re-identified, especially if de-identification is done in a shallow approach (e.g., by removing learners’ full names, emails). The reason is that learners’ personal “footprints” also reside in their artifacts and interaction patterns with the educational platform. Yacobson et al. presented an example where they re-identified the school that the learners were in based on de-identified clickstream logs [67]. Similarly, being teaching software engineering long enough, the paper’s fourth author would argue it is possible to tell learners’ demographic characteristics by reading their code.

The tension between usefulness and anonymity of the data is not likely to be solved by legislation. Hoel et al. analyzed three different privacy frameworks in selected countries [32] and presented clear differences on value focuses – e.g., the European framework focuses on individuals and the Asian privacy framework focuses on the organizations. Though we have observed that the legislation in one region can have an influence on future legislations in other countries [25], the time for those data protection legislations to “converge” (if possible) may take a long time.

As a result, the best short-term result we can have could be a community consensus in the EDM and LA (Learning Analytics) research communities. In addition, when an anonymized dataset is posted/shared, we advocate that researchers limit the *additional information* provided about the student population to reduce the risk of re-identification. For example, a dataset generated by “graduate Algorithms

II students in an R2 university on the east coast of the U.S.” is more likely to be re-identified than a dataset generated by “students in a graduate Algorithms course”.

4.3 A protocol for configuring privacy policies

A common definition for privacy is the POQ framework: “some person or persons P, some domain of information O, and some other person or persons Q, such that P has privacy regarding O with respect to Q” [54]. For example, Alice (P) took an online course with sponsorship from her employer (Q). Her course completion status (O) is accessible by her employer; in other words, P does not have privacy regarding to O with respect to Q. This privacy policy may not be configurable by the learner based on the privacy policy of some online learning platform, for example, edX [16].

Though the POQ framework can serve as a basis for privacy policies, it leaves out some essential components [55]. The privacy protocol helps learners manage privacy in any learning environment. Hoel and Chen suggest the policy should achieve privacy by negotiating “with each student” [31]. Certain components should be added to the POQ framework to extend it for educational service providers.

First, the lifespan of privacy policies should be added. To follow the example above, when Alice leaves the current company, should the former employer still have access to Alice’s records on edX? Besides, the purpose of the planned usage of the data (e.g., to gain generalized knowledge of the student population, to predict individual student’s success in a course, etc.) should be part of the protocol. Educational data can be “justifiably collected and analyzed for one educational context, but not another” [55]. Moreover, privacy protocols should stipulate that learners can access data analysis results based on their data. It is common for researchers to use students’ data to predict student success (or failure) [36]. When there is a prediction, not all the students are willing to see this information, and some educators may not be ready to share this information with students.

5. CONCLUSION

Overall, it does not seem likely that legislation related to educational-data privacy in different countries will be harmonized in the near future. Many datasets from education settings have re-identification risk, even after personal information is removed. Therefore, the research community has to move forward and establish a certain level of consensus to discourage research projects that are of high ethical risk and relatively low research value. Seeking excessive personal data on learners from the social web could be one of them. EDM researchers and third-party tool providers should take responsibility to foster a trusting relationship between learner and teacher, and learner and institution.

Like any survey paper, this work is not specific enough to guide each and every research action, and it will not cover all legislation relevant to an EDM researcher. Within a couple of years, most of this information will be supplanted by new legislation, research paradigms, innovative technologies, and research by the exciting generation of upcoming EDM researchers. Being aware of and vigilant against all possible risks will protect the interests of the EDM research’s most important stakeholders: learners, students and teachers.

6. REFERENCES

- [1] I. F. Anabo, I. Elexpuru-Albizuri, and L. Villardón-Gallego. Revisiting the Belmont Report's ethical principles in internet-mediated research: perspectives from disciplinary associations in the social sciences. *Ethics and Information Technology*, 21(2):137–149, 2019.
- [2] W. M. Association et al. World Medical Association Declaration of Helsinki. Ethical principles for medical research involving human subjects. *Bulletin of the World Health Organization*, 79(4):373, 2001.
- [3] J. T. Avella, M. Kebritchi, S. G. Nunn, and T. Kanai. Learning analytics methods, benefits, and challenges in higher education: A systematic literature review. *Online Learning*, 20(2):13–29, 2016.
- [4] M. Bailey, D. Dittrich, E. Kenneally, and D. Maughan. The Menlo Report. *IEEE Security Privacy*, 10(2):71–75, Mar 2012.
- [5] N. Baym. Social Networks 2.0. *The handbook of Internet studies*, 2:384, 2011.
- [6] 李栋 . 黑客入侵报名信息网站盗取徐玉玉信息 (Hacker attacked registration website and stole Xu Yuyu's information) , Sep 2016.
<http://tc.people.com.cn/n1/2016/0910/c183008-28705847.html>.
- [7] 陈晶晶 . 学生信息不容泄露 (Students' information must not be leaked) , Sep 2018.
<http://m.people.cn/n4/2018/0917/c3521-11620211.html>.
- [8] 陈禹潜 . 泄露学生个人信息没有理由放过 (There is no reason to allow someone who leaks students' personal information to get away without punishment) , Feb 2020.
<http://edu.people.com.cn/gb/n1/2020/0207/c1053-31575345.html>.
- [9] A. Casares. Deep dive into the Edmodo data breach, Oct 2017. <https://medium.com/4iqdelvedeep/deep-dive-into-the-edmodo-data-breach-fl207c415ffb>.
- [10] G. Chen, D. Davis, J. Lin, C. Hauff, and G.-J. Houben. Beyond the MOOC platform: gaining insights about learners from the social web. In *Proceedings of the 8th ACM Conference on Web Science*, pages 15–24, 2016.
- [11] Columbia University. Data Retention: Research, 2020. <https://research.columbia.edu/content/data-retention>.
- [12] A. Constantino. Data breach exposed personal info of nearly 6,000 Montgomery County student accounts | WTOP, 2019. <https://wtop.com/montgomery-county/2019/12/data-breach-exposed-personal-info-of-nearly-6000-montgomery-county-student-accounts/>.
- [13] A. Cormack. Downstream consent: A better legal framework for Big Data. *Journal of Information Rights, Policy and Practice*, 1(1), 2016.
- [14] J. P. Daries, J. Reich, J. Waldo, E. M. Young, J. Whittinghill, A. D. Ho, D. T. Seaton, and I. Chuang. Privacy, anonymity, and big data in the social sciences. *Communications of the ACM*, 57(9):56–63, 2014.
- [15] H. Drachsler and W. Greller. Privacy and Learning Analytics—it's a DELICATE issue. *Proceedings of LAK: International Conference on Learning Analytics & Knowledge*, 16, 2016.
- [16] edX. Notice: On may 15, 2018, edx adopted an amended privacy policy, providing as follows. <https://www.edx.org/edx-privacy-policy>.
- [17] S. Foo. School district officials investigating possible breach of online grading system, Feb 2020. <https://www.kusi.com/school-district-officials-investigating-possible-breach-of-online-grading-system/>.
- [18] L. Forde, J. D'Andrea, and N. Stornebrink. Legal matters: Schools and data privacy, May 2015. <https://www.teachermagazine.com.au/articles/legal-matters-schools-and-data-privacy>.
- [19] P. B. Forsyth, L. L. Barnes, and C. M. Adams. Trust-effectiveness patterns in schools. *Journal of Educational Administration*, 2006.
- [20] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS 15*, Oct 2015.
- [21] R. D. Goddard, M. Tschannen-Moran, and W. K. Hoy. A Multilevel examination of the distribution and effects of teacher trust in students and parents in urban elementary schools. *The Elementary School Journal*, 102(1):3–17, 2001.
<http://www.jstor.org/stable/1002166>.
- [22] A. Godwin, A. R. H. Thielmeyer, J. A. Rohde, D. Verdin, B. S. Benedict, R. A. Baker, and J. Doyle. Using topological data analysis in social science research: unpacking decisions and opportunities for a new method. In *2019 ASEE Annual Conference & Exposition*, Tampa, Florida, June 2019. ASEE Conferences. <https://peer.asee.org/33522>.
- [23] D. Golightly. Google, New Mexico AG Spar Over Chromebook Student Data Collection, 2020. <https://www.androidheadlines.com/2020/02/google-new-mexico-attorney-general-lawsuit-student-data-collection-chromebook.html>.
- [24] P. Golle. Revisiting the uniqueness of simple demographics in the US population. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 77–80, 2006.
- [25] G. Greenleaf. GDPR-Lite and Requiring Strengthening—Submission on the Draft Personal Data Protection Bill to the Ministry of Electronics and Information Technology (India). *UNSW Law Research Paper*, pages 18–83, 2018.
- [26] D. J. Hand. Aspects of data ethics in a changing world: where are we now? *Big Data*, 6(3):176–190, 2018.
- [27] H. Heevner. *FERPA in a Modern World*. PhD thesis, Pennsylvania State University, 2017. https://sites.psu.edu/heevner/files/2017/04/Heevner_FERPA_Final-1fpmggy.pdf.
- [28] B. Herold. Google Under Fire for Data-Mining Student Email Messages, 2014. <https://www.edweek.org/ew/articles/2014/03/13/26google.h33.html>.
- [29] T. Hoel and W. Chen. Implications of the European data protection regulations for learning analytics design. In *Workshop paper presented at the international workshop on learning analytics and*

- educational data mining (LAEDM 2016) in conjunction with the international conference on collaboration technologies (CollabTech 2016), Kanazawa, Japan-September, pages 14–16, 2016.
- [30] T. Hoel and W. Chen. Towards Developing an Educational Maxim for Privacy and Data Protection in Learning Analytics. In *EC-TEL Workshop on Ethics and Privacy for Learning Analytics, Tallinn, Estonia, September*, volume 12, 2017.
 - [31] T. Hoel and W. Chen. Privacy and data protection in learning analytics should be motivated by an educational maxim—towards a proposal. *Research and Practice in Technology Enhanced Learning*, 13(1):20, 2018.
 - [32] T. Hoel, W. Chen, and D. Griffiths. Is international consensus about privacy policies for learning analytics possible? In *Draft workshop paper presented at LAK17 workshop on LA policies*, 2017.
 - [33] L. Irwin. The GDPR: What exactly is personal data?, Jan 2020. <https://www.itgovernance.eu/blog/en/the-gdpr-what-exactly-is-personal-data>.
 - [34] A. C. Ivy and L. Alexander. The Nuremberg Code. *Trials of war criminals before the Nuremberg military tribunals under control council law*, 10:181–182, 1949.
 - [35] D. Kay, N. Korn, and C. Oppenheim. Legal, risk and ethical aspects of analytics in higher education. *Analytics series*, 2012.
 - [36] G. Kennedy, C. Coffrin, P. De Barba, and L. Corrin. Predicting success: how learners’ prior knowledge, skills and activities predict MOOC performance. In *Proceedings of the fifth international conference on learning analytics and knowledge*, pages 136–140, 2015.
 - [37] M. Khalil and M. Ebner. De-identification in learning analytics. *Journal of Learning Analytics*, 3(1):129–138, 2016.
 - [38] R. Koch, M. Golling, and G. D. Rodosek. Geolocation and verification of IP-addresses with specific focus on IPv6. In *Cyberspace safety and security*, pages 151–170. Springer, 2013.
 - [39] S.-J. Lee. The relations between the student–teacher trust relationship and school success in the case of Korean middle schools. *Educational studies*, 33(2):209–216, 2007.
 - [40] C. F. Lynch. Who prophets from big data in education? New insights and new challenges. *Theory and Research in Education*, 15(3):249–271, 2017.
 - [41] K. Mivule. Data Swapping for Private Information Sharing of Web Search Logs. *Procedia Computer Science*, 114:149–158, Sep 2017.
 - [42] A. Narayanan and E. W. Felten. No silver bullet: De-identification still doesn’t work. *White Paper*, pages 1–8, 2014. <https://www.cs.princeton.edu/~arvindn/publications/no-silver-bullet-de-identification.pdf>.
 - [43] A. Narayanan and V. Shmatikov. Robust De-anonymization of Large Sparse Datasets. *2008 IEEE Symposium on Security and Privacy (sp 2008)*, 2008.
 - [44] D. F. Nettleton, A. Orriols-Puig, and A. Fornells. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, 33(4):275–306, 2010.
 - [45] A. A. of Collegiate Registrars and A. Officers. Comparing FERPA and GDPR, Mar 2018. <https://www.aacrao.org/resources/newsletters-blogs/aacrao-connect/article/comparing-ferpa—gdpr>.
 - [46] Office of Research Integrity. Data Management, 2020. https://ori.hhs.gov/education/products/rcradmin/topics/data/tutorial_11.shtml.
 - [47] P. Ohm. Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization. *UCLA Law Review*, 57:1751, Aug 2009.
 - [48] A. Pardo and G. Siemens. Ethical and privacy principles for learning analytics. *British Journal of Educational Technology*, 45(3):438–450, 2014.
 - [49] M. Parry. Harvard researchers accused of breaching students’ privacy. *The chronicle of higher education*, 10, 2011.
 - [50] M. Parsell. Pernicious virtual communities: Identity, polarisation and the Web 2.0. *Ethics and Information Technology*, 10(1):41–56, 2008.
 - [51] P. Prinsloo and S. Slade. Ethics and learning analytics: Charting the (un)charted. In *Handbook of Learning Analytics*. SOLAR, 2017.
 - [52] M. A. Rahman, T. Rahman, R. Laganieri, N. Mohammed, and Y. Wang. Membership inference attack against differentially private deep learning model. *Transactions on Data Privacy*, 11:61–79, Feb 2018. <http://www.tdp.cat/issues16/tdp.a289a17.pdf>.
 - [53] L. S. Romero. Trust, behavior, and high school outcomes. *Journal of Educational Administration*, 2015.
 - [54] A. Rubel and R. Biava. A framework for analyzing and comparing privacy states. *Journal of the Association for Information Science and Technology*, 65(12):2422–2431, 2014.
 - [55] A. Rubel and K. M. Jones. Student privacy in learning analytics: An information ethics perspective. *The information society*, 32(2):143–159, 2016.
 - [56] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes. ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models. *Proceedings 2019 Network and Distributed System Security Symposium*, 2019.
 - [57] N. Sclater, A. Peasgood, and J. Mullan. Learning analytics in higher education. *London: Jisc*. Accessed February, 8(2017):176, 2016.
 - [58] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. *2017 IEEE Symposium on Security and Privacy (SP)*, 2017. https://www.cs.cornell.edu/~shmat/shmat_ak17.pdf.
 - [59] S. Slade and A. Tait. Global guidelines: Ethics in learning analytics. *International Council for Open and Distance Education*, Mar 2019.
 - [60] L. Sweeney. Simple demographics often identify people uniquely. *Health (San Francisco)*, 671:1–34, 2000.
 - [61] S. Turkle. *Life on the Screen*. Simon and Schuster, 2011.
 - [62] United Kingdom Parliament. Children Act 1989, c. 41, 1989.

- <http://www.legislation.gov.uk/ukpga/1989/41/contents>.
- [63] United Kingdom Parliament. Education Act 1996, c. 56, 1996.
<http://www.legislation.gov.uk/ukpga/1996/56/part/IX/chapter/IV>.
 - [64] United Kingdom Parliament. Education Act 2005, c. 18, 2005.
<http://www.legislation.gov.uk/ukpga/2005/18/part/4/crossheading/information>.
 - [65] US Department of Health Human Services et al. The Belmont Report, 1979.
https://videocast.nih.gov/pdf/ohrp_appendix_belmont_report_vol.2.pdf.
 - [66] S. Vallor. Social Networking and Ethics. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2016 edition, 2016.
<https://plato.stanford.edu/archives/win2016/entries/ethics-social-networking/>.
 - [67] E. Yacobson, G. Alexandron, and S. HersHKovitz. De-identification is not enough to guarantee student privacy: De-anonymizing personal information from basic logs. In *Companion Proceedings 10th International Conference on Learning Analytics & Knowledge (LAK20)*, Dec 2019.
 - [68] E. Young. Educational privacy in the online classroom: FERPA, MOOCs, and the big data conundrum. *Harv. JL & Tech.*, 28:549, 2014.
 - [69] M. Zimmer. “But the data is already public”: on the ethics of research in Facebook. *Ethics and information technology*, 12(4):313–325, 2010.

Course Recommendation for University Environments

Boxuan MA
Kyushu University
ma.boxuan.611@s.kyushu-
u.ac.jp

Yuta Taniguchi
Kyushu University
taniguchi@ait.kyushu-u.ac.jp

Shin'ichi Konomi
Kyushu University
konomi@acm.org

ABSTRACT

Recommending courses to students is a fundamental and also challenging issue in the traditional university environment. Not exactly like course recommendation in MOOCs, the selection and recommendation for higher education is a non-trivial task as it depends on many factors that students need to consider. Although many studies on this topic have been proposed, most of them only focus either on historical course enrollment data or on models of predicting course outcomes to give recommendation results, regardless of multiple reasons behind course selection behavior. To address such a challenge, we first conduct a survey to show the underlying characteristic of the course selection of university students. According to the survey results, we propose a hybrid course recommendation framework based on multiple features. Our experimental result illustrates that our method outperforms other approaches. Also, our framework is easier to interpret, scrutinize, and explain than conventional black-box methods for course recommendation.

Keywords

Educational Data Mining; Recommender Systems; University Environments

1. INTRODUCTION

Course selection in university is a crucial and challenging problem that students have to face. It is difficult to decide which courses they should take because there are a large number of courses opened each semester and students have to spend a lot of time exploring those courses. Moreover, the decisions they make shape their future in ways they may not be able to conceive in advance.

We collected a dataset during 2015 and 2018 from our university to gain a better understanding of the elective course enrollment patterns. Figure 1(a) presents the distribution of the enrolled course number of students on the left and the distribution of the popularity for each course of our university on the right. There are hundreds of elective courses offered by the university while averagely students only select a few of them to satisfy the requirements for their degree program. Figure 1(b) shows the distribution of the enrolled courses for each semester. We can also see that students may take courses in the first two years mostly (semester1~semester4), because they may potentially be busy with an internship or finding jobs in the third and fourth year.

Boxuan Ma, Yuta Taniguchi and Shinichi Konomi "Course Recommendation for University Environment" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 460 - 466

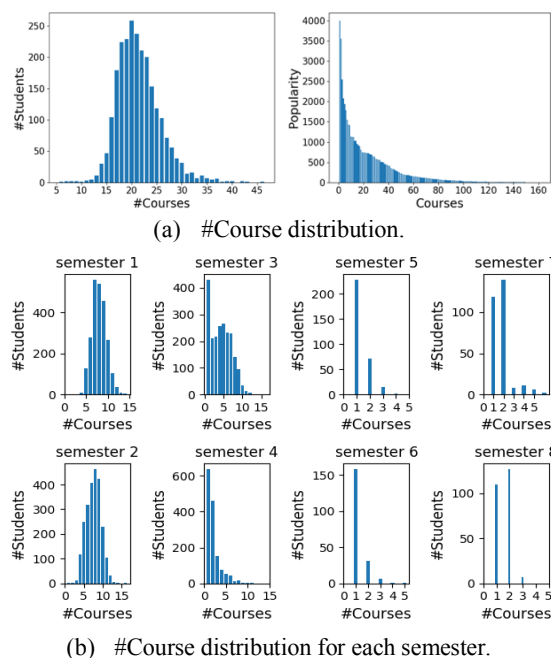


Figure 1. Distribution of courses.

From the discussion above, a safe conclusion could be drawn that due to a large number of available but unfamiliar courses, course selection is a critical activity for students.

With the increasing amount of available data about undergraduate students and their enrollment information, data-driven methods supporting decision making have gained importance to empower student choices and scale advice to large cohorts [14]. Many relevant studies on course recommendation focus on online learning platforms such as MOOCs. Other studies on course recommendation use datasets collected in physical university environments, however, they rely on approaches that are similar to the ones used in recommending MOOC courses without fully considering the different reasons involved in course selection process in physically-based university environments.

In fact, course recommendation for higher education can be more "messy and unorganized" [1] as it depends on many factors that students need to concern. Intuitively, the reasons behind course selection are manifold. Likewise, students who enrolled in the same course may have completely different orientations based on their own reasons, which serves as different criteria for course selection [34]. It inspires us to try to find more useful features for the recommendation.

To make the point clear, a survey is conducted on 81 students in our university to better understand student perceptions and attitudes for their course selection process. [10] Figure 2 shows the main underlying reasons for their course selection.

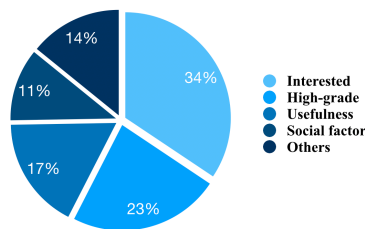


Figure 2. The distribution of main reasons for course selection.

(1) Interest

As Figure 2 shows, the overall most important factor was students' interest and it is often taken as a main contributing factor to the recommendation. However, students may not choose courses based purely on their interest in the university environment. It is expected that students will be more inclined to choose courses that do not require too much effort or difficulty. For example, some students would not enroll in a course which contains contents they are interested in, they just choose the course that allows them to get credits easily.

(2) High-Grade

Improperly selecting courses would seriously affect the students' course achievements, which enforces students to drop out [12]. Getting relatively high grades for students is another factor influences student's choice especially for successful students. Some students even prefer to choose what they perceived would be an easier course for fear that a tougher course might lower their GPA.

(3) Learning goal and career plan

It is natural to recommend courses that align with student's learning goals and career plans as students consider the usefulness of courses as an important factor in their course selection process. However, first-year students may lack learning goals and career planning for the future, and the choice of courses is aimless. Also, student interest and goal can change as they explore and discover something meaningful on and off campus.

(4) Social Aspect

Social factor also plays a part in the course selection process. For example, some students prefer to enroll in a course with their friends or classmates together. Potts et.al [21] conclude that the risk of social isolation is a problem in the learning process especially for first-year students at university, who have difficulty navigating their new academic and environment. Tinto [22] concludes that participation in a collaborative learning group encourages student's attendance and class participation. Therefore, the classmates or friends based social links could be important information in course recommendation.

(5) Popularity

As shown in Figure 1(a), the long-tail distribution of course popularity indicates that students are more motivated to choose popular courses as their first choice. However, the popular courses will be filled up quickly while others will not be selected by students frequently.

In summary, all these discussions above indicate that there are complex constraints and contexts that have to be considered together to balance all those factors above, made more difficult by the multiple objectives that students want to maximize and risks they want to hedge against. For example, choosing challenging courses of value while maintaining a high GPA [16]. This suggests

that recommendations that are aimed only at one or a few factors are likely not enough to help the students.

To address these challenges which have not been well explored in the research community, we propose our hybrid course recommendation framework, which incorporates different criteria in a modular way. Moreover, in our approach selection criteria can further be prioritized by the student. We believe that weaving those criteria could increase the usability of our recommendations compared to previous work focusing only on one of the two. Also, our framework is very efficient and easy to interpret.

2. RELATED WORK

2.1 Course selection

Some work has been done on analyzing the college students' course selection. Morsy and Karypis [23] investigated how the student's academic level when they take different courses, relate to their graduation GPA and time to degree. This study suggests that course recommendation approaches could use this information to better assist students towards academic success, by graduating on-time with high GPA. Also, understanding students' reasons for enrolling in a course provides key information for recommending courses and improving students' learning experiences [24-27].

Additionally, there is still a lack of study on the factors that influence students' course selection in university and how the course selection would impact the students' educational achievement.

2.2 Personalized Course Recommendation

Various approaches have been used in applications for course recommendation by learning from historical enrollment data [32, 33].

Content-based filtering approaches recommend a course to a student by considering the content of the course and clustering course and student into groups to gain similarity between them [2,3]. Collaborative filtering approaches recommend a course to a student by investigating student's similarity with the student's historical data in a system and predict the course that the student would be interested in [4-6]. Association rules based on frequent patterns are used to discover interesting relations that describe previous course selections from students [8,9]. Recently, other methods including sequence discovery and representation learning have been used in this domain [11,19,20]. However, those systems often behave like a "black box", i.e., recommendations are presented to the users, but the rationale for selecting recommendations is often not explained to end-users.

2.3 Grade Prediction

While some researchers have focused on between-course enrollment data, others have focused on models of predicting grades in future courses [13-6]. Based on what courses they previously took and how well they performed in them, the predicted grades give an estimation of how well students are prepared for future courses, then recommending courses to students that will help them to get relatively high grades [18,28,29,31].

However, these methods can be prone to recommending relatively easier courses in which students usually get high grades [17]. In addition, there are some students who like challenge difficult courses if they are interested in or think it is helpful for their future career, for those students, the grade prediction based recommendations are not enough.

Despite the significant success of various course recommendations, constraints on the number of student preferences in the university environment resulting in inflexibility where a student's requirements do not align perfectly with those built into the system. In contrast to the aforementioned approaches, our model combines the concerns of performance and interest together. Also, it has the benefit of allowing for a custom weighting of those components, as well as the increased explanatory value of the model itself.

3. PROPOSED METHOD

We first give the definition of our recommendation problem in Section 3.1. Then we propose our hybrid course recommendation framework with three subsections introducing our Interest-based Score, Timing-based Score, and Grade-based Score in detail. Finally, those different scores are used in our course recommendation algorithm introduced in Section 3.3.

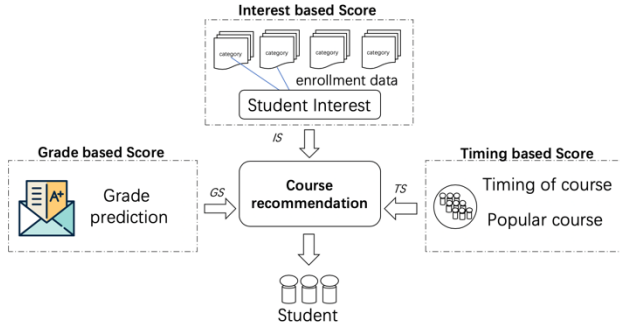


Figure 3. Overview of the proposed course recommendation.

3.1 Problem Formulation

Like every classic recommendation task, there are two basic elements *user* and *item* in our course recommendation task, where a user represents a student and an item represents a course. We use S to denote a set of students and C to denote a set of courses. Each $s \in S$ has enrolled some courses denoted by $C_s \subseteq C$ and each $c \in C$ has its enrollment set denoted by $S_c \subseteq S$. Let T denote a set of all available semesters, and t to denote a specific semester. Generally, there are 8 semesters for 4 academic years degree program. Let G denote a set of grades that student could get, and each $g \in G$ denote a specific grade that student obtained for a course. Let $E = \{(s, c, g, t) | s \in S, c \in C_s, g \in G, t \in T\}$ be the set of all enrollment relations, which means student s enrolled in course c in semester t , and got the final grade g .

Given enough students enrollment data (S, C, E) , our goal is recommending courses to a specific student s which are not in C_s for next semester.

3.2 Framework

According to the result of the survey shown in Section 1, students may concern different factors while they choose courses. Inspired by that, we propose our hybrid course recommendation framework that considers student interest, the timing of taking the course and the predicted grade of the student together. Figure 3 shows the overview of the proposed course recommendation.

For each pair of student and course (s, c) , we need to understand how suitable the course is for the specific student. We use three different aspects to calculate the $Score(s, c)$ for each pair of student and course:

(i) *Interest based Score (IS)*, which is to measure how interesting the course is for a specific student. (ii)

Timing based Score (TS), which is to measure how suitable students enroll in the course at a specific time (semester) since different courses may have different suitable time periods. (iii) *Grade based Score (GS)*, which is to predict students' performance for the course.

We propose our approaches to estimate IS , TS and GS , respectively. Then, they are fused by a student-specific weight parameter as the $Score(s, c, t)$. Once all of the $Score(s, c, t)$ have been computed, the k courses with the highest score are selected.

3.2.1 Interest-based Score

Let s and c be a student and a course, respectively, the goal of interest score estimation is to calculate $IS(s, c)$.

In our framework, we extract user interest from student historical enrollment behaviors. Since each course of university belongs to a category, let $CATE$ denote the set of all categories, $cate$ to denote a specific category, then $CATE = \{cate_1, cate_2, \dots, cate_{|CATE|}\}$. We think that there is a strong relationship between student interest and course categories. For instance, a student frequently enrolls in courses which belong to "Computer Science" may imply that the student has an interest in this category or he may have personal learning goal in this domain. Hence, it is appropriate to recommend the student the courses such as "Python Programming" and "Data science".

For a student s , the idea is to count the number of courses that he enrolled in and belongs to a category, i.e., $Num(s, cate)$. Then, all of the values are normalized as the preference score from 0 to 1, denoted as $p(s, cate)$, which is defined as equation (1).

$$p(s, cate) = \frac{Num(s, cate)}{\max_{cate' \in CATE} (Num(s, cate'))} \quad (1)$$

For a student s , the preference vector \mathbf{P}_s , is obtained by the preference score of each category, which is defined as equation (2).

$$\mathbf{P}_s = (p(s, cate_1), p(s, cate_2), \dots, p(s, cate_{|CATE|})) \quad (2)$$

We can further use \mathbf{P}_s to calculate the similarity between student s and other students. Let s_i and s_j be two students, the similarity between s_i and s_j can be measured by the cosine similarity measurement as shown below.

$$sim(s_i, s_j) = \frac{\mathbf{p}_{s_i}^T \cdot \mathbf{p}_{s_j}}{\|\mathbf{p}_{s_i}\| \times \|\mathbf{p}_{s_j}\|} \quad (3)$$

For the convenience of computation, we use a matrix form representation $P = (\mathbf{p}'_{s_1}; \mathbf{p}'_{s_2}; \dots; \mathbf{p}'_{s_{|S|}})$ to denote the interest of all students where $\mathbf{p}'_s = \mathbf{p}_s / \|\mathbf{p}_s\|$ means the normalization of \mathbf{p}_s . Then the similarity matrix Sim can be simply written as:

$$Sim = P^T \times P \quad (4)$$

where $Sim_{i,j}$ is the result of $sim(s_i, s_j)$.

Based on the similarity, we could estimate the user-based interest score. For a student s and a course c , the Interest Score denoted as $IS(s, c)$, is defined as (5), where $S_{s,k}$ indicates the set of top- k similar students of s as neighbors, and $I_{C_{s'}}$ is an indicator function whose value is 1 when $c \in C_{s'}$.

$$IS(s, c) = \frac{\sum_{s' \in S_{s,k}} I_{C_{s'}} \times sim(s, s')}{\sum_{s' \in S_{s,k}} sim(s, s')} \quad (5)$$

Furthermore, we try to utilize students' major information together with their similarity as equation (6).

$$sim_m(s, s') = \lambda sim(s, s') + (1 - \lambda) samemajor(s, s') \quad (6)$$

Where $samemajor(s, s')$ function equal to 1 if student s and student s' have the same major, otherwise, the function equal to 0. λ (limited from 0 to 1) is used to control the weight between similarity and major information. The underlying rationale is that each major has its owner preference on courses enrolling, students have the same major will generally make a similar choice in course selection. Also, students in the same major are more likely to be friends or classmates, which brings potential social link information into the course recommendation. Then equation (5) can be rewritten as below.

$$IS(s, c) = \frac{\sum_{s' \in S_{s,k}} I_{c,s'} \times sim_m(s, s')}{\sum_{s' \in S_{s,k}} sim_m(s, s')} \quad (7)$$

3.2.2 Timing and popularity based Score

Different courses may have different suitable time periods (semesters). For example, in each department, courses can be taken by students of different grades, e.g., freshman or sophomore. Previous studies showed that the timing of courses has a strong correlation with student graduation GPA and time to degree [23]. Based on that, we assume that the timing of courses is also important for course selection. The suitable timing of courses will help students for good grades and successful graduation in a timely manner.

For each course c , we define the *Timing based Scores (TS)*, denoted as $TS(c, t)$, where t indicates a specific semester. In our framework, TS is considered from two aspects:

(1) Which semester is more suitable for taking this course? For a specific course, we sum up the number of enrollments for every semester and normalize all of the values. The result is denoted as $T_t(c, t)$.

$$T_t(c, t) = \frac{Num(c, t)}{Max_{t' \in T}(Num(c, t'))} \quad (8)$$

where $Num(c, t)$ represents the number of enrollments of course c in semester t , and T indicates the set of all time periods, i.e., 8 semesters for 4 academic years degree program.

(2) Which courses are popular now? For a specific semester, we sum up the number of enrollments for every course and normalize all of the values. The result is denoted as $T_p(c, t)$.

$$T_p(c, t) = \frac{Num(c, t)}{Max_{c' \in C}(Num(c', t))} \quad (9)$$

where C indicates the set of all courses. $T_t(c, t)$ and $T_p(c, t)$ are then fused by the harmonic mean since we want both of the two values are relatively high. The final *Timing based Score*, $TS(c, t)$ can be defined as:

$$TS(c, t) = 2 \times \frac{T_t(c, t) \times T_p(c, t)}{T_t(c, t) + T_p(c, t)} \quad (10)$$

Therefore, we can use $TS(c, t)$ to ensure that the semester t is suitable for taking the course c and the course c is suitable for taking in the semester t .

3.2.3 Grade-based Score

Improperly selecting courses would seriously affect the students' course achievements, which may decrease their GPA even enforce students to drop out. Accurately predicting students' grades in future courses has attracted much attention as it can help identify at-risk students early [30].

We use the grade prediction method called cross-user-domain collaborative filtering proposed by Ling et al. [12]. For predicting the score of each course $c \in C$ for each student $s \in S$, a small set

of senior students who have already enrolled on course c and have the most similar previous score distribution to student s will be discovered by means of Pearson correlation coefficient. The underlying rationale is that students with similar scores in the previous courses will generally obtain similar scores in the subsequent courses.

Let S_s denote the set of senior students who have already enrolled on course c . For any senior student $s_s \in S_s$, the following Pearson correlation coefficient is used to measure the course score similarity between student s and the senior student s_s .

$$sim(s, s_s) = \frac{\sum_{i \in C_{ss}} (g_{si} - \bar{g}_{si}) (g_{s_s i} - \bar{g}_{s_s i})}{\sqrt{\sum_{i \in C_{uu}} (g_{si} - \bar{g}_{si})^2} \sqrt{\sum_{i \in C_{ss}} (g_{s_s i} - \bar{g}_{s_s i})^2}} \quad (11)$$

where C_{ss} denotes the courses that are enrolled by both students s and s_s , $g_{s,i}$ and $g_{s_s,i}$ denote the grade of course i by students s and s_s respectively. \bar{g}_{si} and $\bar{g}_{s_s i}$ denote the average grade of courses enrolled by students s and s_s , respectively. Accordingly, the grade of the course c by student s can be predicted as follows.

$$g_{sc} = \frac{\sum_{s_s \in S_{s,k}} (g_{s_s c} - \bar{g}_{s_s c}) \times sim(s, s_s)}{\sum_{s_s \in S_{s,k}} sim(s, s_s)} \quad (12)$$

where $S_{s,k}$ indicates the set of top- k similar senior students of s . It should be noticed that students often achieve inconsistent grades in the various courses they take, and different students may have varying grades deviations, i.e. the grades deviation compared with the average grades among all students. Similarly, different courses may have varying grades deviations, i.e. the score deviation compared with the average score among all courses. In order to deal with those variations. We use the grade deviation of student s and the grade deviation of course c to predict student grades. Accordingly, equation (12) could be rewritten as below.

$$g_{sc} = b_{sc} + \frac{\sum_{s_s \in S_{s,k}} (g_{s_s c} - b_{s_s c}) \times sim(s, s_s)}{\sum_{s_s \in S_{s,k}} sim(s, s_s)} \quad (13)$$

where $b_{sc} = \mu + b_s + b_c$ denotes the baseline estimate for g_{sc} with μ being the overall mean grade of all courses enrolled by all students, $b_s = \bar{g}_s - \mu$ being the grade deviation of student s and $b_c = \bar{g}_c - \mu$ being the grade deviation of course c , where \bar{g}_s is the overall mean grade of student s and \bar{g}_c is the overall mean grade of course c .

Finally, we could use the grades that students are expected to obtain in future courses to boost the performance of our recommendation. The final *Grade based Scores*, $GS(s, c)$ can be defined as normalized values of grades.

$$GS(s, c) = \frac{g_{sc}}{Max_{c' \in C}(g_{s c'})} \quad (14)$$

The total score of student and course pair *score* (s, c) can be written as:

$$Score = \alpha \times IS(s, c) + \beta \times TS(s, t) + \gamma \times GS(s, c) \quad (15)$$

Where α, β, γ are parameters to control the proportion of weights from different sources. By taking those scores into account simultaneously, a course that the student interested in, and suitable for him to take to get a high grade could be ranked higher than other courses. Also, student could control the weighting of those components to have a better understanding of the data and decision-making.

3.3 Course Recommendation Algorithm

The whole framework can be written as Algorithm 1. Student set S , course set C , enrollment set $E(s, c, g, t)$ are input and output is a list of recommendations R_s for student's next semester that includes up to k recommendations per student.

Algorithm 1: Generating a list of course recommendations for student

Input :

Student set S , course set C , enrollment set $E(s, c, g, t)$;

Output :

Recommendation results for each student R_s .

- 1 Calculate student interest p_s for each $s \in S$ by equation 1;
 - 2 Calculate student interest similarity by equation 4;
 - 3 Calculate student grade similarity by equation 11;
 - 4 Calculate student deviation of each student;
 - 5 Calculate course grade deviation of each course;
 - 6 **foreach** $s \in S$ **do**
 - 7 Calculate user interest-based score $IS(s, c)$ by equation 7;
 - 8 Calculate timing-based score $TS(c, t)$ by equation 10;
 - 9 Calculate user grade-based score $GS(s, c)$ by equation 13 and equation 14;
 - 10 Calculate final score $Score(s, c, t)$ by equation 15;
 - 11 Let R_s be the sorted list of C ordered by its $Score(s, c, t)$ in descending order.
 - 12 **endfor**
-

4. EVALUATION

In this section, we conducted a series of experiments to evaluate the effectiveness of our proposed method. We first describe the dataset and experimental settings. Next, the evaluation methodology and metrics are introduced in detail. Finally, the results are shown in Section 4.4.

4.1 Dataset

This work focuses on undergraduate students in a traditional educational institution. We used a dataset from our university that spans for 5 years. The dataset consisted of per-semester course enrollment information of 2,366 students from 12 departments, with a total of 38,968 pseudonymized enrollment records from 2014 through 2018. Each row of the course enrollment data contained semester and department information, an anonymous student ID and course information included course name, instructor and course category.

4.2 Experiment Settings

4.2.1 Data selection

The most natural approach to evaluate the model is to split the data by semesters. As shown in Figure 1(b), most of the undergraduate students may take courses in the first two years. Therefore, for students who enrolled in 2015, the semester of Spring 2015 was used for training, the subsequent semesters of Fall 2015, Spring 2016 and Fall 2016 are regarded as the testing semesters, each of which is tested separately. The results are evaluated by comparing the predicted courses and the ground-truth courses he/she has enrolled in.

4.2.2 Comparison

We name our methods as Hybrid Course Recommendation (HCR). We compare our method with two group popularity approaches [14] and Random recommendation (Random). The two group popularity approaches including the department level (Grp-Pop-1), which recommend the most popular courses in the major, and the academic level (Grp-Pop-2), which recommend the most popular courses on the major and the academic level of the student ("freshmen", "sophomores", "juniors", and "seniors").

4.3 Evaluation Metrics

Like previous work [11,14,15,20], we used Recall@ns and Coverage as the evaluation metric for the performance.

Coverage is measured based on the percentage of courses that have been recommended at least once to students, which describes the ability of a recommendation system to explore the long-tail item.

Recall@ns is the percentage of actually enrolled courses of s in semester t that were contained in the recommendation list, where ns is the number of courses that the student took in the target semester. The reported metrics are averaged out across all students. Since our proposed course recommendation method considers both student interest and the grade he/she may obtain, we cannot only use the Recall metric, and instead, we use a variation of it. For the list of the courses R_s that recommended to a student s , Let T_s is the set of courses in the test set of s , A_s is the set of courses which student is expected to get the grade equal to or higher than his/her average previous grade. We use the ratio of $|R_s \cap T_s \cap A_s|$ and $|T_s \cap A_s|$ to measures the fraction of the actual well performed courses that are retrieved.

4.4 Results

4.4.1 Interest-based Score

In collaborative filtering strategy, taking how many similar students as neighbors is an important problem which is sensitive to the quality of the result. We investigate the performance of our interest model with different neighbor numbers. As shown in Figure 4, the performance of the model increases with the increase of neighbor number at first then decreases. According to the observation above, we pick a practical value 40 as the value of the neighbor number parameter in our follow-up experiments.

We also investigate the performance of our interest model with different weights between similarity and major information. As shown in Figure 5, we can observe that the performance of the model increases with the decrease of λ in terms of Recall. The reason is that the model considers not only the similarity but also the major information. That is, each major has its own preference for courses enrolling, major information will improve the performance of the algorithm.

However, 100% recall could be bad because the system just recommends what students do anyway. We noticed that the Coverage also decreases with the decrease of λ . The model seems benefit from the major information while scarifying the diversity of results. Recommendations for courses at other departments sometimes are useful to mine more long-tail student interest while students usually ignored that these courses existed or that their content matched their interests. To achieve the best performance of recommendation, we need to make a trade-off. According to the observation above, we set λ as 0.2 in our follow-up experiments.

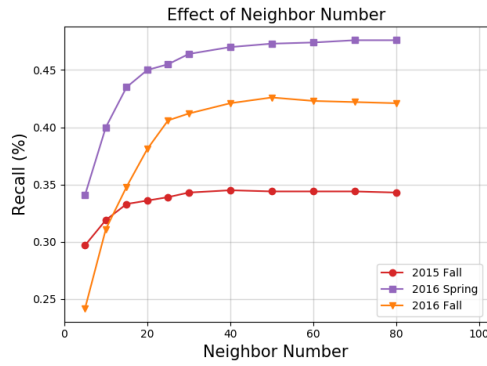


Figure 4. Performance of different neighbor numbers.

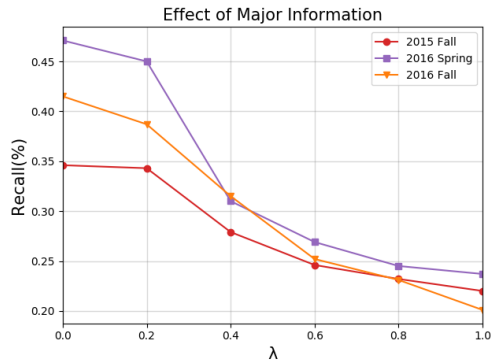


Figure 5. Evaluation of major information.

4.4.2 Influence of different factors

To illustrate the influence of different factors, we set each parameter α, β, γ from 0 to 1 with a step size 0.05 to find the optimal combination.

Table 2. Performance of different parameters

Model	$(\alpha, \beta, \gamma) = (1, 0, 0)$	$(\alpha, \beta, \gamma) = (0, 1, 0)$	$(\alpha, \beta, \gamma) = (0, 0, 1)$	$(\alpha, \beta, \gamma)^*$
Recall@ns	0.380	0.412	0.341	0.434
Recall(a)@ns	0.311	0.315	0.332	0.322
Coverage	0.534	0.212	0.356	0.516

As shown in Table 2, the interest score and timing score has a good explanatory value for the recommendation than others. Also, the suitable timing of taking a course will help students to get a good grade. A recommendation only based on the grade has a good performance for recommending high grade courses. However, the results cannot help all the students. We reached the best Recall@ns with $(\alpha = 0.4, \beta = 0.45 \text{ and } \gamma = 0.15)^*$.

The results indicate that recommendations that are aimed only at one factor are likely not to be satisfied by every student. As we discussed before, different students may have completely different orientations based on their own reasons, which serves as different criteria such as their preferences, interests, needs, performance, etc. Such a hybrid system could provide explanations and user

controls for different categories of target students to support the interpretation of the data and decision-making.

Table 3. Evaluation of course recommendation

Semester	Model	Recall@ns	Recall(a)@ns	Coverage
Fall 2015	Random	0.048	0.036	-
Fall 2015	Grp-Pop-1	0.374	0.306	0.272
Fall 2015	Grp-Pop-2	0.452	0.342	0.342
Fall 2015	HCR	0.472	0.393	0.578
Spring 2016	Random	0.025	0.020	-
Spring 2016	Grp-Pop-1	0.325	0.201	0.305
Spring 2016	Grp-Pop-2	0.423	0.372	0.237
Spring 2016	HCR	0.431	0.402	0.342
Fall 2016	Random	0.002	0.002	-
Fall 2016	Grp-Pop-1	0.326	0.243	0.213
Fall 2016	Grp-Pop-2	0.441	0.387	0.250
Fall 2016	HCR	0.463	0.392	0.559

4.4.3 Comparison result

We analyze the performance of different algorithms. The results in Table 3 show that our framework performs well when compared with other methods.

As the results show, both of the Recall and Recall(a) of Random recommendation strategies are very low since there are a large number of courses, but each student only averagely chooses a few courses per semester. Hence, it is difficult to recommend the right course. Popularity approaches are having considerably satisfactory performance in Recall since popular courses which are taken by students frequently usually attract most of students. However, Grp-Pop-1 and Grp-Pop-2 do not consider student preference, it is also difficult to mine more long-tail student interest as the Coverage is low. In addition, Grp-Pop-1 and Grp-Pop-2 are not good in Recall(a) since they only consider the popular courses, ignore the performance the student is expected to get in the recommended courses.

5. CONCLUSION

This research aims to recommend suitable courses for learners and study how to design a personalized course recommendation in the university environments. In this paper, we propose a hybrid course recommendation framework that considers student interest, the timing and popularity of courses, and predicted performance of students, simultaneously. Experiments are conducted to confirm the effectiveness of the proposed approach. The results show that the proposed hybrid course recommendation approach performed well compared to other methods. Also, the model itself is flexible in the sense that one can easily adjust or extend it by changing the recommendation formula and incorporate more information.

6. REFERENCES

- [1] E. Babad and A. Tayeb. 2003. Experimental analysis of students' course selection. *British Journal of Educational Psychology*, 73(3):373–393.
- [2] Piao G, Breslin JG. 2016. Analyzing MOOC Entries of Professionals on LinkedIn for User Modeling and Personalized

- MOOC Recommendations. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. ACM, 291–292.
- [3] Apaza RG, Cervantes EV, Quispe LC, Luna JO. 2014. Online Courses Recommendation based on LDA. In *SIMBig*. 42–48.
- [4] BYDŽOVSKÁ, Hana. 2016. Course Enrollment Recommender System. *Proceedings of the 9th International Conference on Educational Data Mining*. Raleigh, NC, USA: International Educational Data Mining Society. 312–317.
- [5] Khorasani ES, Zhenge Z, Champaign J. 2016. A Markov Chain Collaborative Filtering Model for Course Enrollment Recommendations: 2016 IEEE International Conference on Big Data (Big Data). 3484 – 3490.
- [6] Jing, X., Tang, J. 2017. Guess you like: course recommendation in Moocs. In: *Proceedings of the International Conference on Web Intelligence*, ACM, 783–789.
- [7] Bhumichitr K, Channarukul S, Saejiem N, Jiamthapthaksin R, Nongpong K. 2017. Recommender Systems for university elective course recommendation. *14th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (2017): 1–5.
- [8] Aher SB, Lobo LM. 2013. Combination of machine learning algorithms for recommendation of courses in E-Learning system based on historical data. *Knowledge-Based Systems* (2013), 1–14.
- [9] Bendakir N, Aïmeur E. 2006. Using association rules for course recommendation. *Proceedings of the AAAI Workshop on Educational Data Mining*. Vol. 3.
- [10] Ma, B.X., Lu, M., Taniguchi, Y. and Konomi, S. 2020. Exploring the Design Space for Explainable Course Recommendation Systems in University Environments. In *Companion Proceedings of the 10th International Conference on Learning Analytics & Knowledge*.
- [11] Polyzou A, Nikolakopoulos AN, Karypis G. 2019. Scholars Walk: A Markov Chain Framework for Course Recommendation. *International Educational Data Mining Society*. (2019 Jul).
- [12] Huang L, Wang CD, Chao HY, Lai JH, Philip SY. A score prediction approach for optional course recommendation via cross-user-domain collaborative filtering. *IEEE Access*. (2019 Feb 7);7:19550–63.
- [13] Sweeney, M., Lester, J., Rangwala, H., and Johri, A. 2016. Next-term student performance prediction: A recommender systems approach. *Journal of Educational Data Mining* 8, 1, 22–51.
- [14] Elbadrawy, A. and Karypis, G. 2016. Domain-aware grade prediction and top-n course recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 183–190.
- [15] Morsy, S. and Karypis, G. 2017. Cumulative knowledge-based regression models for next-term grade prediction. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 552–560.
- [16] Jiang W, Pardos ZA and Wei Q. 2019. Goal-based course recommendation. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*. ACM, 36–45.
- [17] Morsy S, Karypis G. 2019. Will this Course Increase or Decrease Your GPA? Towards Grade-aware Course Recommendation[J]. arXiv preprint arXiv:1904.11798.
- [18] Okubo F, Yamashita T, Shimada A, Ogata H. 2017. A neural network approach for students' performance prediction. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference* (2017 Mar 13). 598–599.
- [19] Pardos ZA, Jiang W. 2019. Combating the Filter Bubble: Designing for Serendipity in a University Course Recommendation System. arXiv preprint arXiv:1907.01591. 2019.
- [20] Pardos ZA, Fan Z, Jiang W. 2019. Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. *User Modeling and User-Adapted Interaction*. (2019 Apr 1). 487–525.
- [21] Potts BA, Khosravi H, Reidsema C, Bakharia A, Belonogoff M, Fleming M. 2018. Reciprocal peer recommendation for learning purposes. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge* (2018 Mar 7). 226–235.
- [22] Tinto V. 1997. Classrooms as communities: Exploring the educational character of student persistence. *The Journal of higher education*. (1997 Nov 1).68(6):599–623.
- [23] Morsy, S. and Karypis, G. 2019. A study on curriculum planning and its relationship with graduation gpa and time to degree. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*. ACM, 26–35.
- [24] Kinnunen P, Malmi L. 2006. Why students drop out CS1 course? In *Proceedings of the second international workshop on Computing education research* (2006 Sep 9). 97–108.
- [25] Crues R, Bosch N, Anderson CJ, Perry M, Bhat S, Shaik N. 2018. Who They Are and What They Want: Understanding the Reasons for MOOC Enrollment. *International Educational Data Mining Society*. (2018 Jul 16).
- [26] Kardan, H. Sadeghi, S. S. Ghidary, and M. R. F. Sani. 2013. Prediction of student course selection in online higher education institutes using neural network. *Computers & Education*. vol. 65. 1–11.
- [27] Feng W, Tang J, Liu T X. 2019. Understanding dropouts in MOOCs. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2019. 33: 517–524.
- [28] Elbadrawy, A., Studham, R. S., and Karypis, G. 2015. Collaborative multi-regression models for predicting students' performance in course activities. In *Proceedings of the 5th International Learning Analytics and Knowledge Conference*.
- [29] Hu, Q. and Rangwala, H. 2018. Course-specific markovian models for grade prediction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 29–41.
- [30] Hu, Qian and Huzefa Rangwala. 2019. Reliable Deep Grade Prediction with Uncertainty Estimation. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*. ACM, 76–85.
- [31] Backenköhler M, Scherzinger F, Singla A, Wolf V. 2018. Data-Driven Approach towards a Personalized Curriculum. *International Educational Data Mining Society*.
- [32] Morsomme R, Alferez SV. 2019. Content-based course recommender system for liberal arts education. In *Proceedings of the 12th International Conference on Educational Data Mining* 2019. Vol. 748, p. 753.
- [33] Bydžovská H. 2016. Course Enrollment Recommender System. *International Educational Data Mining Society*.
- [34] Esteban A, Zafra A, Romero C. 2018. A Hybrid Multi-Criteria Approach Using a Genetic Algorithm for Recommending Courses to University Students. *International Educational Data Mining Society*. (2018 Jul).

Methodology of measure of similarity in student video sequence of interactions.

Boniface Mbouzaou
Polytechnique Montréal

boniface.mbouzaou@polymtl.ca

Michel C. Desmarais
Polytechnique Montréal
michel.desmarais@polymtl.ca

Ian Shrier
McGill University
ian.shrier@mcgill.ca

ABSTRACT

Massive online Open Courses (MOOCs) make extensive use of videos. Students interact with them by pausing, seeking forward or backward, replaying segments, etc. We can reasonably assume that students have different patterns of video interactions, but it remains hard to compare student video interactions. Some methods were developed, such as Markov Chain and Edit Distance. However, these methods have caveats as we show with prototypical examples. This paper proposes a new methodology of comparing video sequences of interaction based both on time spent in each state and the succession of states by computing the distance between the transition matrices of the video interaction sequences. Results show the proposed methodology can better characterize video interaction in a task to discriminate which student is interacting with a video, or which video a student is interacting with.

Keywords

MOOC, Distance matrix, Edit Distance, Markov Chain, Optimal Matching Distance

1. INTRODUCTION

In online learning contexts, learner engagement is often measured by their interaction with video. The simplest measure is the total amount of time spent on video listening that can be used as an engagement measure [6]. But the availability of detailed interactions with a video allows more sophisticated measures, and comparison between video interactions.

Two common methods used to find the similarity between video interactions are the Markov Chain and Edit distance measures. The main limitation of using Markov Chain to compare video interactions sequences is that state transition probabilities do not take into account the time between states. Many sequences can have the same transitions probability matrix but represent different styles and length.

Boniface Mbouzaou, Michel Desmarais and Ian Shrier "Methodology to measure of similarity in student video sequence of interactions." In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 467 - 472

By contrast, the Edit distance approach to comparing video interaction sequences may take time into account if the sequences of events are mapped to a time scale and represented as activity segments, such as in [4]. However, large offset, such as a pause, in similar activity sequences will create large Edit distances that will shadow the similarity.

A methodology that can simultaneously take into account the time and transitions between activities could help the analysis of video interaction. It could help the analysis of the MOOCs and online teaching systems learning in video intensive environments, and could help to extract meaningful patterns of video interactions. It has often been used to classify students to identify students at risk (see for eg. [14, 8, 2]).

2. BACKGROUND

Among the different techniques to analyze video clickstream, some focus on extracting patterns, or motif, between events [3, 17, 16]. Descriptive statistics such as the video proportion played are also commonly used (see for eg. [15]). However, our focus is on measuring distance, or conversely similarity between video interaction patterns, and what are the most useful representations for that purpose.

We review the basics of the two families of methods and representations used in measuring video interaction similarity in more details and discuss their issues, before describing previous work with each approach, and then describe and evaluate the proposed method.

First, we describe the event data and a common transformation of events into activity sequences.

2.1 Events and activity sequences

Data on student interaction with videos relies on the notion of events associated to timestamps, such as "play" at 0:00:00 and "pause" at 0:00:10. There are five basic video interaction events: (1) *load*, (2) *play*, (3) *pause*, (4) *seek* and (5) *stop*.

The student can be considered in a state of listening to a video between 0 sec. and 10 sec., and in pause state thereafter. For example, suppose we have two students interactions:

Interaction sequences:

- 1: Play (4 seconds) then Pause (4 seconds) and then Play (4 seconds),

- 2: Pause (2 seconds) then Play (8 seconds) and then Pause (2 seconds).

Each student spent 12 seconds in total interaction with video. We can transform those two patterns of interaction into a sequence of activity states of 1 second intervals:

Activity sequences:

- 1: P1-P1-P1-P1-Pa-Pa-Pa-Pa-P1-P1-P1-P1
2: Pa-Pa-P1-P1-P1-P1-P1-P1-Pa-Pa-Pa
(P1=Play and Pa=Pause)

We will name this type of sequence an *activity sequence*, where a polling interval is defined and the activity corresponds to the last event that occurred. Activity sequence encoding has been used in a few studies of student interaction patterns with a learning system [4, 1].

We now turn to how these sequences can be represented.

2.2 Markov Chain representation

A Markov chain is specified by a set of states and transitions between states. The process starts in one of the state s_i , then moves to another s_{i+1} with a probability of $p_{i,i+1}$. The Markov property stipulates that the transition probability is independent of states prior to s_i .

Considering the video interaction events as states, a student interaction can be represented as a Markov state transition matrix, where cells contain frequencies of transitions in the sequence, normalized such that row sums are 1, and thus represent transition probabilities.

For example, the two *interaction sequences* in the section above would result in the following event sequences:

Event sequences:

- 1: P1-Pa-P1
2: Pa-P1-Pa

Contrary to *activity sequences*, *event sequences* do not carry the notion of a polling at regular time interval and ignore the time stamps on events. These event sequences would in turn result in a Markov Chain that is common to both:

$$\mathbf{M}_{\text{seq1.1}} = \mathbf{M}_{\text{seq2.1}} = \begin{matrix} & \begin{matrix} \text{play} & \text{pause} \end{matrix} \\ \begin{matrix} \text{play} \\ \text{pause} \end{matrix} & \begin{pmatrix} 0/1 & 1/1 \\ 1/1 & 0/1 \end{pmatrix} \end{matrix}$$

A measure of distance between sequences can be computed from the two Markov matrices, such as the Frobenius norm of the cell-wise difference between the matrices. More on this below.

The limit of using Markov Chain to compare video event sequences lies in the fact that transitions probabilities can be the same for very different sequences. This issue is evident in the two sequences above that end up having the same Markov transition matrix. While it can be alleviated by having a start end state, it is clear that the loss of state duration information will lead to a loss of valuable information.

However, Markov Chains are efficient at capturing transition patterns and have been used with some success for clustering [12, 11], for creating student profiles of interactions [10, 5], and for simulated students [7].

2.3 Sequence Edit Distance method

The sequence Edit Distance method relies on measures found with word distances, where alphabet similarity between words is the basis of calculating similarity.

Edit Distance (ED), generates distances that represent the minimal cost in terms of insertions, deletions and substitutions for transforming one sequence to another. The cost of each deletion, insertion or insertion is 1 by default. This algorithm was originally proposed by Levenshtein [9] and is most common when computing distances between words [13]. For video listening sequences, the principle is the same but the alphabet is represented by the activity. For example, the ED measure for activity sequences 1 and 2 above yields a distance of 9 over a maximum of 12.

A notable property of the ED measure is that sequences of different lengths will necessarily have a non null distance, and therefore potentially miss regularities in interaction patterns of different length sequences. On the contrary, a Markov Chain representation is not sensitive to sequence length, or to the number of transitions for that matter (since the row sums are all normalized to 1), whilst its capacity to capture interaction patterns in sequences of different length.

3. PROPOSED METHOD, TMED

The proposed method, named TMED, is a combination of the two techniques: the Markov Chain and the ED measure. The combination of results give a full similarity between each pair of student sequences of interactions benefiting of advantages from both techniques.

3.1 Transition matrix

The video transition matrix of a student s for a video is expressed as:

$$\mathbf{M}_s = \begin{matrix} & \begin{matrix} \text{load} & \text{play} & \text{pause} & \text{seek} & \text{stop} \end{matrix} \\ \begin{matrix} \text{load} \\ \text{play} \\ \text{pause} \\ \text{seek} \\ \text{stop} \end{matrix} & \begin{pmatrix} m_{1.1} & m_{1.2} & m_{1.3} & m_{1.4} & m_{1.5} \\ m_{2.1} & m_{2.2} & m_{2.3} & m_{2.4} & m_{2.5} \\ m_{3.1} & m_{3.2} & m_{3.3} & m_{3.4} & m_{3.5} \\ m_{4.1} & m_{4.2} & m_{4.3} & m_{4.4} & m_{4.5} \\ m_{5.1} & m_{5.2} & m_{5.3} & m_{5.4} & m_{5.5} \end{pmatrix} \end{matrix}$$

where $m_{j,k}$ is the *number of transitions* from event j to event k in an *activity sequence* obtained from an *interaction sequence*. And \mathbf{M}_s is the transition matrix of student s interacting with a video. Contrary to a Markov Chain, rows do not necessarily sum to 1. In the case where no event occurs and the student remains in the same state for awhile (playing video or pausing video, for eg.) the increase of the matrix element m_i is the maximum number of transitions possibles within the time spent in that state counting the transition from one state to the same state.

3.2 Distance between two transition matrices

The distance between two student transition matrix is expressed as:

$$\begin{aligned} d(\mathbf{M}_{s1}, \mathbf{M}_{s2}) &= \|\mathbf{M}_{s1} - \mathbf{M}_{s2}\|_F \\ &= \sqrt{\sum_{i=1}^5 \sum_{j=1}^5 (m_{s1,j} - m_{s2,j})^2} \end{aligned}$$

An important question is what is the polling interval to choose. This interval will determine the total number of transitions in M_s . The choice is determined by the minimal interval required to avoid skipping events while transforming the event sequence to the activity sequence. In our case, this interval is set to 3 per second and it applies to all video interactions. The total number of transitions, $T_{s,i}$ in a given interaction matrix \mathbf{M} for sequence s and video i is therefore:

$$T_{s,i} = L_{s,i} * N \quad (1)$$

where $L_{s,i}$ is the length of the interaction time and N is the polling interval.

The similarity between two interaction video sequences based on transition matrices with a video is then expressed as:

$$S_{mat}(\mathbf{M}_{s1}, \mathbf{M}_{s2}) = 1 - Dis(\mathbf{M}_{s1}, \mathbf{M}_{s2}) \quad (2)$$

$$Dis(\mathbf{M}_{s1}, \mathbf{M}_{s2}) = \frac{d(\mathbf{M}_{s1}, \mathbf{M}_{s2})}{T_{s1} + T_{s2}} \quad (3)$$

Where $S_{mat}(\mathbf{M}_{s1}, \mathbf{M}_{s2})$ is the similarity level between sequence of interaction of student $s1$ and student $s2$ of video i using matrix of interactions and $Dis(\mathbf{M}_{s1}, \mathbf{M}_{s2})$ is the dissimilarity between them. $d(\mathbf{M}_{s1}, \mathbf{M}_{s2})$ is the distance among them. T_{s1} and T_{s2} are the number of transitions of student $s1$ and student $s2$ sequence of the video i . If $S_{mat}(\mathbf{M}_{s1}, \mathbf{M}_{s2})$ is 0 then the two sequences are completely dissimilar and when it is 1 then they are completely similar. Between 0 and 1 shows the percentage of similarity between the two sequences of transitions.

3.3 Edit Distance measure (ED)

For each pair of sequences, we compute the ED distance to obtain the distance matrix and from there compute the level of similarity among them. The level of similarity between two sequences is computed using ED distance as:

$$S_{om}(seq_{s1}, seq_{s2}) = 1 - \frac{dist_{om}(seq_{s1}, seq_{s2})}{\max(T_{s1}, T_{s2})} \quad (4)$$

Where $S_{om}(seq_{s1}, seq_{s2})$ is the similarity level between sequence of student $s1$ and sequence of student $s2$ of video i and $dist_{om}(seq_{s1}, seq_{s2})$ is the ED distance between the two sequences and T_{s1} and T_{s2} are the numbers of transition of the sequence of each student given in equation (1). $\max(T_{s1}, T_{s2})$ is the maximum between the number of transitions of the two student sequences of interactions.

3.4 TMED

The last step of this proposed methodology is to combine the two techniques by taking for each pair of sequences the proper level of similarity among the levels given by each technique. This is meant to take into account the complementary of those techniques: one can find styles and give good similarity for sequences of different lengths and the other gives regularity among sequences and gives good similarity among sequences from the same range length. The final similarity level is then given by:

$$S(seq_{s1}, seq_{s2}) = Select(S_{om}(seq_{s1}, seq_{s2}), S_{mat}(\mathbf{M}_{s1}, \mathbf{M}_{s2})) \quad (5)$$

Where $S(seq_{s1}, seq_{s2})$ is the level of similarity between sequence of interaction $s1$ and $s2$, $S_{om}(seq_{s1}, seq_{s2})$ similarity

level between the two sequences based on ED distance as expressed in equation (4) and $S_{mat}(\mathbf{M}_{s1}, \mathbf{M}_{s2})$ similarity level between the two sequences based on sequence matrix as expressed in equation (2).

The function *Select()* selects S_{mat} similarity if one of the two sequences is less than the half-length of the other, and selects the maximum level of similarity between the proposed method and the ED method otherwise.

One takes the maximum between ED similarity and matrix similarity to avoid the ED drawback of finding dissimilarity between sequences of same range of length but some mismatch between states as illustrated in section 4 below. The flow of the proposed method is illustrated in Figure 1 from the sequences to the computation of their similarity level.

4. VALIDATION

To validate the proposed method, we compare its capacity of finding the level of similarity between sequences with existing methods, namely the Markov Chain technique as used by Klingler et al.[8] and the ED based method used for clustering the same kind of sequences of interactions.

4.1 Prototypical cases

We first test the approach over prototypical cases where the patterns are obvious to the eye. For this purpose we take two main cases: sequences of same lengths of transitions and sequences of different length of transitions. For the same sequence length interactions, we considered a cyclic same sequence of transitions as illustrated in Figure 2a. The cycle of transitions is: Lo-P1-Pa-P1-Pa-Se-P1-St. The cycle of transition can start anywhere and finish by *St* for any of the sequence.

The expected level of similarity should be close 100% as it is the same sequence following a cycle. The result based on ED distance cannot find that level of similarity as shown in Figure 2b compared to the Markov based method in Figure 2c (with some exceptions which do not reach the 100% similarity as expected, but close enough to be considered as such) and the proposed method in Figure 2d (finds perfect match of style by 100% similarity in each case). For these cyclic sequences, the proposed method and the Markov based similarity methods are performing better than ED based method in finding similarity between two cyclic same sequences of interactions.

The second validation of the proposed method is to compare it to a Markov based method for different length sequences given known similarities. For this purpose, we considered four sequences of same transitions levels as shown in Figure 3a. In this case, the percentage of transition between states is the same, but the time spent in each state is different from one sequence to another. The expected level of similarity depends here on the lengths of each sequence as the succession of states are the same for all four sequences. We should have then as result a progressive increase in level of similarity from the shortest sequence to the longest.

The result from the Markov Chain based method as in Figure 3a could not find the different levels of similarity as the percentage of transition between the states is preserved with

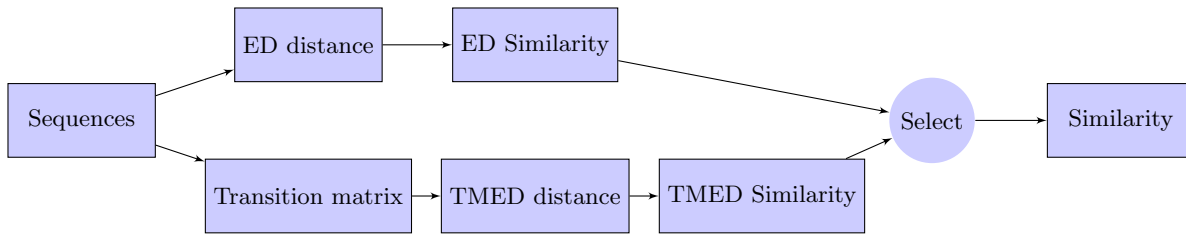


Figure 1: Flow of the proposed method to compute similarity between students' video sequences. "Select" is the selection process between the two technique similarity.

different sequences lengths. The proposed method performs better as shown in Figure 3b because it is based on the number of transitions rather than probability of transition as Markov Chain is.

4.2 Real dataset

The experiment on a real set of video interaction logs aims to test and compare the ability of the proposed method to recognize (1) the student behind an interaction log (data contains 4800 students), and (2) the video behind an interaction log. While this task is of no practical use, since both the video and student associated with an interaction log are already known in general, it provides a ground truth dataset to assess the discrimination power of each approach.

We choose three well-known classifiers such as support vector machine (SVM), boosted tree (GBM) and K-nearest neighbor (KNN) for each method of representation of sequence of interactions to predict first the student and then video to which sequence of interaction belongs. If a specific representation of student sequence of interaction is predictable in terms of which video and student that interact with the video, that means that the representation is able to better distinguish different types of interaction and even showing the specificity of a video in the way that students interact with it.

For the first part of the experiment where we predict student to which the sequence representation belongs, the algorithm arbitrarily selects one sequence of each student to predict among the nine (9) same student sequences representation and trains on the eight (8) others student sequences representation. The matrix distance used for Markov chain sequence representation and the proposed TMED representation is the one described above in section 2.2. In these two cases the dimension of the representation of each sequence is 25, that represent the 25 elements of transition matrix of each sequence representation as described in section 3.3. For the OM sequence representation, the matrix distance used is the one described in section ?? above. For the prediction 80% of the data is use for training and 20% for prediction. Each experiment is repeated 400 times using different set of students to predict (from 3 to 15 students). The data set is organized in such that all the student sequence present in the data set selected, the training set has 8 of their sequence representation and one in the testing set in each prediction run.

In the second part of the experiment, we used the same representations of student footage but instead of predicting the

student, we predicted the video the student interacted with. We used the same training (80% of the data) and test (20% of the data) sets, making sure that in the data we had the same number of students interacting with each video. Since each student has nine (9) sequences of interaction representation, the number of predicted classes (video 1 to video 9) in each data set considered is the same regardless of the number of students considered. For this reason, balanced precision was included in the results to avoid the effect of having more students. Again, in this case, at each prediction run, the algorithm ensures that each student sequence representation in the data set considered is the same as its sequence representations in the test set in each run.

4.3 Real data results

The results show that the proposed TMED method through the level of similarity. Through the tests of validation on prototypical data, the proposed method yields better results than the other two existing methods as one can see through Figures 2 and 3. For the same sequence represented as a cyclic sequence of interaction with various ways of representation show in Figure 2 (a) the expected degree of similarity 100% but only the proposed method give us the closest results to the expected one as shown in Figure 2. One can also see in this figure that the Markov chain based similarity is the second-best estimation of similarity after the proposed method based one.

When we consider a same sequence of states with different lengths of time as shown in Figure 2 (a), the expected results of similarity is a progressive increase of level of similarity according to the length of the sequence. The classic Markov chain based method could not find that the length of sequences are different whereas TMED method is able to find it well (Figure 3 (c)).

The experiment over the real data tasks tests the capacity of each method of representation of video interaction to identify each sequence of interaction in terms of student and video sequences. Results show better accuracy for TMED than the other ones (table 1). The performance parameters on student prediction using SVM, GBM and KNN on predicting five (5) students and twelve (12) students with nine (9) records of each student (where eight (8) records are for training and predicting one record of each student).

For predicting video, the complete results for forty-five (45) records from five (5) different students and hundred and eight (108) records from twelve (12) students in predicting the nine (9) videos are shown in table 2. They demonstrate

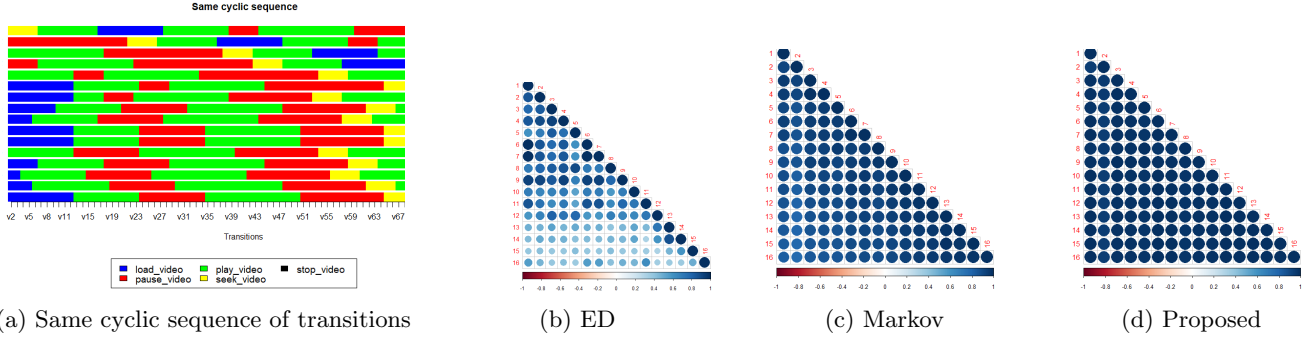


Figure 2: Result of similarity: (a) The cycle starts and follows the same pattern of transition to close the cycle (b) Similarity based on Edit Distance (ED) cannot recognize the similarity of cyclic sequences. (c) Similarity based on Markov Chain can recognize the similarity, with some exceptions that not reach 100%. (d) Proposed TMED similarity can recognize cyclic sequences.

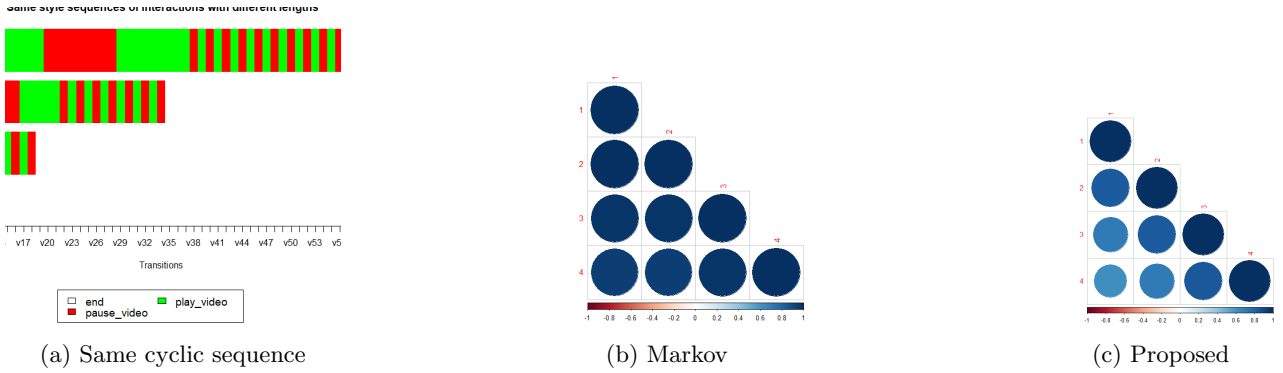


Figure 3: Similarity results from the sequence in (a): (b) similarity based on Markov Chain cannot recognize the duration in each state.(c) proposed TMED similarity can recognize the fact that those sequences are same but the level of similarity is based on the time spent in each state.

Predictions: 45 records, 5 target students									
Approach:	SVM			GBM			KNN		
Method:	ED	MC	TMED	ED	MC	TMED	ED	MC	TMED
Accuracy	0.60	0.00	0.80	0.40	0.00	1.00	0.20	0.22	1.00
F_1	0.75	0.00	0.89	0.57	0.00	1.00	0.33	0.36	1.00
Predictions: 108 records, 12 target students									
Accuracy	0.58	0.18	0.67	0.42	0.36	0.42	0.11	0.00	0.40
F_1	0.73	0.20	0.78	0.59	0.50	0.63	0.20	0.00	0.67

Table 1: Results of Twenty fold cross validation 400 runs of student prediction of 5 and 12 students using three different methods of representation of student interaction with videos showing that the proposed representation technique is performing better than others.

that the proposed method is also better on recognizing both video and student than the two other methods of presentation of student interaction with video.

These results suggest that the proposed method has a better way of representing a student video interaction with videos and so can be used for comparing two different interactions with video.

5. CONCLUSION

The proposed methodology aims to fill out a methodological gap on representing and comparing video sequences of interaction methods. The proposed method overcomes the drawbacks of the previous methods based on Markov Chain and sequence of interactions known as Edit Distance (ED). The main contribution of this proposed method is the fact that it takes into account the time spent in each state and the general style of succession of states. This offers a new tool to researchers who want to compared video viewers interaction and find eventually video style of interaction.

Predictions:		45 records, 9 target videos								
Approach:	Method:	SVM			GBM			KNN		
		ED	MC	TMED	ED	MC	TMED	ED	MC	TMED
Accuracy		0.11	0.33	0.56	0.33	0.56	0.56	0.22	0.22	0.33
F_1		0.20	0.50	0.72	0.50	0.36	0.72	0.36	0.36	0.50
Predictions:		108 records, 9 target videos								
Accuracy		0.22	0.11	0.56	0.11	0.33	0.56	0.22	0.11	0.22
F_1		0.36	0.20	0.61	0.20	0.50	0.61	0.36	0.20	0.36

Table 2: Results of Twenty fold cross validation 400 runs of video prediction using three different methods of representation of student interaction with videos, ED (Edit Distance), MC (Markov Chain), TMED.

TMED combines two styles of representation of video sequence of interaction and computes the similarity based on the advantage of each style of representation. The ED based similarity is generally good on same range length of interaction sequences and the matrix of interaction based representation does better on sequences of different range of length.

The proposed method is also able to better represent a sequence of interaction when doing classification tasks as the results show. In fact, proposed method has a better performance in predicting student sequence of interaction and prediction video when having a representation of a video sequence of interaction.

References

- [1] Yoav Bergner, Zhan Shu, and Alina von Davier. Visualization and confirmatory clustering of sequence data from a simulation-based assessment task. In *Educational Data Mining 2014*, 2014.
- [2] Mina Shirvani Boroujeni and Pierre Dillenbourg. Discovery and temporal analysis of latent study patterns in MOOC interaction sequences. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, pages 206–215. ACM, 2018.
- [3] Christopher G Brinton, Swapna Buccapatnam, Mung Chiang, and H Vincent Poor. Mining MOOC clickstreams: Video-watching behavior vs. in-video quiz performance. *IEEE Transactions on Signal Processing*, 64(14):3677–3692, 2016.
- [4] Michel Desmarais and François Lemieux. Clustering and visualizing study state sequences. In *Educational Data Mining 2013*, 2013.
- [5] Louis Faucon, Lukasz Kidzinski, and Pierre Dillenbourg. Semi-markov model for simulating MOOC students. *International Educational Data Mining Society*, 2016.
- [6] Philip J Guo, Juho Kim, and Rob Rubin. How video production affects student engagement: An empirical study of MOOC videos. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 41–50. ACM, 2014.
- [7] Robert Hlavatý and Ludmila Dömeová. Students’ progress throughout examination process as a markov chain. *International Education Studies*, 7(12):20–29, 2014.
- [8] Severin Klingler, Tanja Käser, Barbara Solenthaler, and Markus H Gross. Temporally coherent clustering of student data. In *EDM*, pages 102–109, 2016.
- [9] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.
- [10] Alice Marques and Orlando Belo. Discovering student web usage profiles using markov chains. *Electronic Journal of e-Learning*, 9(1):63–74, 2011.
- [11] Sylvain Mongy, Fatma Bouali, and Chabane Djeraba. Analyzing user’s behavior on a video database. In *Multimedia data mining and knowledge discovery*, pages 458–471. Springer, 2007.
- [12] Sylvain Mongy, Chabane Djeraba, and Dan A Simovici. On clustering users’ behaviors in video sessions. In *DMIN*, pages 99–103, 2007.
- [13] Gonzalo Navarro. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88, 2001.
- [14] Nirmal Patel, Collin Sellman, and Derek Lomas. Mining frequent learning pathways from a large educational dataset. *arXiv preprint arXiv:1705.11125*, 2017.
- [15] Tanmay Sinha. ” your click decides your fate”: Leveraging clickstream patterns from mooc videos to infer students’ information processing & attrition behavior. *arXiv preprint arXiv:1407.7143*, 2014.
- [16] Tanmay Sinha, Patrick Jermann, Nan Li, and Pierre Dillenbourg. Your click decides your fate: Inferring information processing and attrition behavior from mooc video clickstream interactions. *arXiv preprint arXiv:1407.7131*, 2014.
- [17] Tanmay Sinha, Nan Li, Patrick Jermann, and Pierre Dillenbourg. Capturing “attrition intensifying” structural traits from didactic interaction sequences of MOOC learners. *arXiv preprint arXiv:1409.5887*, 2014.

PIPE: Predicting Logical Programming Errors in Programming Exercises

Dezhuang Miao
School of Data Science and
Engineering
East China Normal University
Shanghai, China
{51185100025,51195100031}@stu.ecnu.edu.cn

Yu Dong
School of Data Science and
Engineering
East China Normal University
Shanghai, China

Xuesong Lu^{*}
School of Data Science and
Engineering
East China Normal University
Shanghai, China
xslu@dase.ecnu.edu.cn

ABSTRACT

In colleges, programming is increasingly becoming a general education course of almost all STEM majors as well as some art majors, resulting in an emerging demand for scalable programming education. To support scalable education, teaching activities such as grading and feedback have to be automated. Recently, online judge systems have been extensively used for programming training, because they are able to automatically evaluate the correctness of programs in real time and thereby make grading work scalable. However, existing online judge systems lack of the ability to give effective feedback on logical programming errors. As such, instructors and teaching assistants are still overwhelmed by the work of helping students fix programs, especially for those novice students. To tackle the challenge, we develop **PIPE**, a deep learning model that is able to **Predict logical Programming Errors** in student programs. The model seamlessly integrates a representation learning model for obtaining the latent feature of a program and a multi-label classification model for predicting the error types in the program, thereby allowing end-to-end learning and prediction. We use the C programs submitted in our online judge system to train PIPE, and demonstrate its superior performance over the baseline models. We use PIPE to implement the error-feedback feature in our online judge system and enable automated feedback on logical programming errors to the students.

Keywords

Online Judge System, Scalable Programming Training, Logical Programming Error, Automated Error Feedback, Deep Learning

1. INTRODUCTION

^{*}Xuesong Lu is the corresponding author.

Dezhuang Miao, Yu Dong and Xuesong Lu "PIPE: Predicting Logical Programming Errors in Programming Exercises" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 473 - 479

The evolution of big data and AI technologies has made programming a ubiquitous skill in almost all industries and thereby led to a massive demand for programming professionals. In colleges and MOOC platforms, programming is no longer a professional course of ICT-related majors and becoming a general education course for all STEM majors and even some art majors. As such there is an urgent need for scalable programming teaching methodologies and learning tools to cater for the increasingly overwhelmed teaching workload. One of the most important mechanisms to achieve scalable teaching is automation. For example, online judge (OJ) systems [22], which are originally used for competitive programming contests, have now been extensively used in programming training mainly due to their ability of automated program evaluation. Given a programming exercise and a set of predefined input, the judge system evaluates a submitted program¹ by comparing the expected output with the actual output obtained from the execution of the program. Such a pair of predefined input and output is called a *test case*. This feature can largely reduce the grading workload of instructors and teaching assistants, and thus make class sizes scalable to some extent.

Despite the ability of automated program evaluation, existing OJ systems often provide to students next-to-zero feedback on programming errors when they submit an incorrect program. We refer to an "incorrect program" as a piece of code that is compilable but generates wrong output for the test cases. The errors in such a program are often termed as "logical errors", as opposed to "common errors" that are related to the use of incorrect syntax. In our teaching, we observe that the students can easily fix common errors with the help of an IDE, but are quite struggling when dealing with logical errors. In the latter case, existing OJ systems only show to students feedback such as "Wrong Answer" and "Runtime Error", and cannot provide any information on detailed types of errors. The problem is even severer in case of a quiz, where students are not allowed to check the test cases². As such, students, especially novices, rely heavily on instructors and teaching assistants to help them fix logical errors, which prevents programming training from becoming more scalable. This has motivated us to develop an automated tool for logical error feedback.

¹Below we use the term 'program' and 'code' interchangeably.

²Otherwise, students may fake the output.

In this work, we develop a deep learning model, PIPE, that is able to predict the detailed types of logical errors and therefore can be deployed in OJ systems to enable automated error feedback. We collect the C programs that are compilable but fail to pass the test cases, submitted by the students in our OJ system. We manually label the programs with a set of predefined types of logical errors. Since each program may contain multiple types of logical errors, we regard the prediction task as a multi-label classification problem. Therefore, the architecture of PIPE is inspired by the work of *code2vec* [2] and *C2AE* [24], which are originally developed to predict semantic properties of code snippets and boost the performance of multi-label classification tasks, respectively. In particular, we first use the idea of *code2vec* to obtain the latent representations of C programs. For each program, in addition to just embedding the code itself, we embed two more types of information in the model input, namely, the corresponding exercise identity and the evaluation results returned by the judge system. Then following the idea in *C2AE*, we also transform the corresponding error types into latent representations with an encoder, and jointly learn deep latent spaces together with the representations of the C programs. The error types are finally reconstructed from the deep latent spaces using a decoder, which are then used to compute the loss function with the true error labels for backpropagation. Thanks to the seamless integration of *code2vec* and *C2AE*, PIPE allows end-to-end training and prediction. We then conduct extensive experiments to demonstrate PIPE's superior performance over the baseline models. We deploy PIPE in our OJ system and show the usage of the automated error-feedback feature.

The rest of the paper is organized as follows. Section 2 presents the detailed architecture of PIPE. Then Section 3 describes the real dataset used in our experiments and presents the performance evaluation of the proposed model. Section 4 gives a brief literature review of related work, and finally Section 5 concludes the work and points out some future work to improve the feature of automated error feedback.

2. THE PIPE MODEL

We describe the architecture and the optimization method of PIPE in this section.

2.1 Architecture Overview of PIPE

Since each program may contain more than one logical error, we regard the error prediction task as a multi-label classification problem. We use the structure of the *C2AE* model as the backbone of PIPE. The *C2AE* model performs joint input and output embedding which correlates the features and the labels, and hence achieves the new state-of-the-art performance on multi-label classification tasks. In particular, PIPE uses a feature mapping \mathbf{F}_x to transform the programs \mathbf{X} and uses an encoding function \mathbf{F}_e to transform the corresponding labels \mathbf{Y} of the logical errors into deep latent spaces \mathbf{L} . Then it utilizes Deep Canonical Correlation Analysis [3] (DCCA) to learn \mathbf{L} for joint program and label embedding. Finally, PIPE uses a decoding function \mathbf{F}_d to recover the label outputs from \mathbf{L} , where \mathbf{F}_e and \mathbf{F}_d thus compose an autoencoder for the reconstruction of the labels. The objective function of PIPE is formulated as follows:

$$\Theta = \min_{\mathbf{F}_x, \mathbf{F}_e, \mathbf{F}_d} \Phi(\mathbf{F}_x, \mathbf{F}_e) + \alpha \Gamma(\mathbf{F}_e, \mathbf{F}_d) \quad (1)$$

where Θ represents the total loss of PIPE, $\Phi(\mathbf{F}_x, \mathbf{F}_e)$ and $\Gamma(\mathbf{F}_e, \mathbf{F}_d)$ denote the loss at the latent space layer for associating features and labels, and the loss at the output layer for reconstructing the labels, respectively. The hyperparameter α balances the two components of the objective function. Once the training is completed, PIPE can throw away the component pertaining to \mathbf{F}_e and use $\mathbf{F}_d(\mathbf{F}_x)$ to predict the logical errors in each program.

In PIPE, we simply use a fully-connected network to implement the functions \mathbf{F}_e and \mathbf{F}_d , respectively. We further leverage the idea in *code2vec* to implement the feature mapping \mathbf{F}_x for program representation learning, as shown in the part surrounded by the red dotted line in Figure 1. Rather than directly embed the source code, *code2vec* first decomposes the program into a collection of paths in its abstract syntax tree (AST) and then learns to aggregate the paths into a single program vector. The method is proved to better capture the regularities that reflect common program patterns and lower the learning effort, compared to learning over original program text. To capture more information about the logical errors pertaining to each particular exercise, we embed the exercise identity and the evaluation results on the test cases returned by the judge system, and concatenate them with the program vector to form a unified feature vector, which we call *program embedding*. Then the program embeddings are transformed into the latent space \mathbf{L} using a fully-connected layer. The architecture overview of PIPE is shown in Figure 1.

2.2 Program Embedding

Firstly, we need to transform the programs into vectorized representations. Following the method in previous work [2, 19], we compile each C program X and parse it to construct an AST. An AST is a tree representation of the abstract syntactic structure of source code, where the nodes denote the various elements appearing in the original source code. By traversing between the AST leaves, we can obtain multiple syntactic paths that represent the context of the corresponding C program. Then the syntactic paths are converted into context vectors, which are used as one type of input to learn the values of program embedding. Each context vector $\mathbf{c}_i \in \mathbb{R}^{3d}$ is concatenated to using three individual vectors, as depicted in Equation 2,

$$\mathbf{c}_i = [\mathbf{s}_i, \mathbf{p}_i, \mathbf{t}_i] \quad (2)$$

where $\mathbf{s}_i \in \mathbb{R}^d$, $\mathbf{p}_i \in \mathbb{R}^d$ and $\mathbf{t}_i \in \mathbb{R}^d$ are the vectorized representation of the source node, the path and the target node of the corresponding syntactic path, respectively. Then each context vector $\mathbf{c}_i \in \mathbb{R}^{3d}$ is transformed into a combined context vector $\hat{\mathbf{c}}_i \in \mathbb{R}^d$ using a shared fully-connected layer, and finally all the combined context vectors are aggregated into a single program vector $\mathbf{v}_p \in \mathbb{R}^d$ using the following attention mechanism,

$$\mathbf{v}_p = \sum_{i=1}^n \alpha_i \cdot \hat{\mathbf{c}}_i \quad (3)$$

$$s.t. \quad \alpha_i = \frac{\exp(\hat{\mathbf{c}}_i^T \cdot \mathbf{a})}{\sum_{j=1}^n \exp(\hat{\mathbf{c}}_j^T \cdot \mathbf{a})}$$

where \mathbf{a} is the attention vector, α_i is the attention weight and n is the number of combined context vectors. The attention vector learns the importance of each combined context

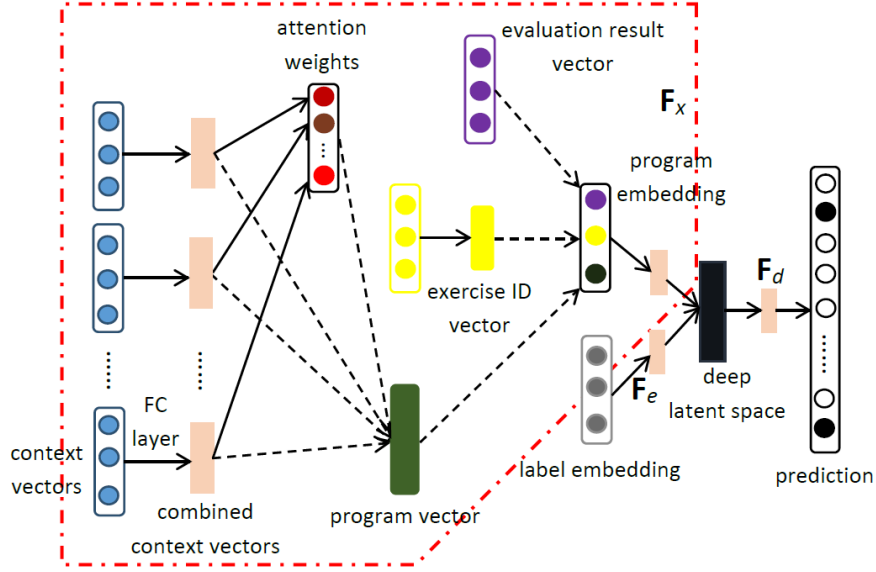


Figure 1: The overview of PIPE's architecture.

vector. To capture more information pertaining to the errors in the program w.r.t a particular exercise, we further embed the exercise identity and the evaluation results on the test cases into two vectors $\mathbf{v}_e \in \mathbb{R}^{d_e}$ and $\mathbf{v}_r \in \{0, 1\}^{d_r}$, respectively. \mathbf{v}_r is a bit vector where 1 indicates a correct output on the test case and 0 otherwise. The vector \mathbf{v}_e restricts the exercise-related characteristics such as functions and algorithmic logic, and the vector \mathbf{v}_r captures specific types of errors since similar logical errors should result in wrong output on similar test cases. Eventually the program embedding \mathbf{v}_x is obtained by concatenating \mathbf{v}_p , \mathbf{v}_e and \mathbf{v}_r , as formulated in Equation 4.

$$\mathbf{v}_x = [\mathbf{v}_p, \mathbf{v}_e, \mathbf{v}_r] \quad (4)$$

2.3 Learning Deep Latent Spaces for Joint Program & Label Embedding

Following the idea in the work [24], we learn deep latent spaces \mathbf{L} to associate program embedding and label embedding, using Deep Canonical Correlation Analysis [7, 3] (DCCA). For each C program X , we simply represent its label Y as a bit vector $\mathbf{v}_Y \in \{0, 1\}^N$, where N is the number of logical error types. The vector \mathbf{v}_Y may contain multiple 1s since each program may have multiple types of logical errors. Then both the program embedding \mathbf{v}_x and the label embedding \mathbf{v}_Y are transformed into a latent vector of size l using a fully-connected layer with the tanh activation function. The holistic functions that mapping X and Y are refer to as \mathbf{F}_x and \mathbf{F}_e , respectively, as depicted in Section 2.1. Then the objective function for correlating the latent representations are formulated as Equation 5,

$$\begin{aligned} \Phi(\mathbf{F}_x, \mathbf{F}_e) &= \|\mathbf{F}_x(\mathbf{X}) - \mathbf{F}_e(\mathbf{Y})\|_F^2 \\ \text{s.t. } \mathbf{F}_x(\mathbf{X})\mathbf{F}_x(\mathbf{X})^T &= \mathbf{F}_e(\mathbf{Y})\mathbf{F}_e(\mathbf{Y})^T = \mathbf{I}, \end{aligned} \quad (5)$$

where $\mathbf{I} \in \mathbb{R}^{l \times l}$ is the identity matrix. By solving the objective function, we enforce the deep latent space \mathbf{L} to associate the programs \mathbf{X} and the labels \mathbf{Y} , and hence $\mathbf{F}_x(X)$ can be used as the input to predicting the label Y .

2.4 Recovering Label Outputs from the Deep Latent Space

In the training phase, the output label \hat{Y} is reconstructed from the latent representation $\mathbf{F}_e(Y)$ using a decoder \mathbf{F}_d , which is simply implemented as a fully-connected layer in this work. In the original work [24], the model uses a label-correlation aware function to calculate the label reconstruction loss $\Gamma(\mathbf{F}_e, \mathbf{F}_d)$ at the output layer, in order to better preserve the label co-occurrence information for multi-label classification task. However, we notice that there is no strong correlation between the labels of the logical error types in our dataset. Hence, we instead use the multi-label cross-entropy function to calculate the label reconstruction loss, as depicted in Equation 6,

$$\begin{aligned} \Gamma(\mathbf{F}_e, \mathbf{F}_d) &= \frac{1}{|\mathbf{Y}|} \sum_{j=1}^{|\mathbf{Y}|} E_j \\ E_j &= - \sum_{i=1}^N y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \end{aligned} \quad (6)$$

where $|\mathbf{Y}|$ is the number of training instances, N is the number of logical error types and y_i equals to 1 if the program contains the corresponding error and equals to 0 otherwise. We use the Sigmoid activation function in the output layer. By solving the loss function, we enforce the autoencoder $\mathbf{F}_d(\mathbf{F}_e(Y))$ to reconstruct the label of the logical error types. Since the latent representation $\mathbf{F}_e(Y)$ and $\mathbf{F}_x(X)$ are highly correlated after the training is completed, $\mathbf{F}_d(\mathbf{F}_x(X))$ can be used to predict the error types of a given C program X .

2.5 Optimization

The gradient of the label-reconstruction loss $\Gamma(\mathbf{F}_e, \mathbf{F}_d)$ can be easily calculated since it is a cross-entropy function. Following the method in [24], the gradient of the association aware loss $\Phi(\mathbf{F}_x, \mathbf{F}_e)$ in the latent space can be calculated with the help of Lagrange multipliers [20]. In particular,

$\Phi(\mathbf{F}_x, \mathbf{F}_e)$ is first reformulated as

$$\Phi(\mathbf{F}_x, \mathbf{F}_e) = \text{Tr}(\mathbf{C}_1^T \mathbf{C}_1) + \lambda \text{Tr}(\mathbf{C}_2^T \mathbf{C}_2 + \mathbf{C}_3^T \mathbf{C}_3), \quad (7)$$

where

$$\begin{aligned} \mathbf{C}_1 &= \mathbf{F}_x(\mathbf{X}) - \mathbf{F}_e(\mathbf{Y}) \\ \mathbf{C}_2 &= \mathbf{F}_x(\mathbf{X})\mathbf{F}_x(\mathbf{X})^T - \mathbf{I} \\ \mathbf{C}_3 &= \mathbf{F}_e(\mathbf{Y})\mathbf{F}_e(\mathbf{Y})^T - \mathbf{I}. \end{aligned}$$

We fix λ to 0.5 in accordance with [24]. Then the gradient w.r.t $\mathbf{F}_x(\mathbf{X})$ and $\mathbf{F}_e(\mathbf{Y})$ can be calculated as

$$\begin{aligned} \frac{\partial \Phi(\mathbf{F}_x, \mathbf{F}_e)}{\partial \mathbf{F}_x(\mathbf{X})} &= 2\mathbf{C}_1 + 4\lambda \mathbf{F}_x(\mathbf{X})\mathbf{C}_2 \\ \frac{\partial \Phi(\mathbf{F}_x, \mathbf{F}_e)}{\partial \mathbf{F}_e(\mathbf{Y})} &= 2\mathbf{C}_1 + 4\lambda \mathbf{F}_e(\mathbf{Y})\mathbf{C}_3. \end{aligned} \quad (8)$$

3. PERFORMANCE EVALUATION

In this section, we evaluate the performance of PIPE on a real dataset collected in our OJ system, and report the experimental results by comparing PIPE with other baseline methods. In the end, we demonstrate an example of the error-feedback feature implemented with PIPE in our OJ system.

3.1 The Dataset and Settings

The real dataset is collected from an introductory C programming course for undergraduate students in our school. The course uses heavily an OJ system to train the students, and we collect all the programs with logical errors submitted by the 29 enrolled students throughout one entire semester. Most programs have less than 50 lines. After cleaning work such as removing repeated submissions of programs with minor changes, we obtain 5196 C programs pertaining to 200 programming exercises. We have carefully designed for each exercise 10 test cases. Then we randomly disseminate the URLs of these programs to 17 senior students and ask them to annotate the labels of logical errors. In order to guarantee the correctness of annotation, they are allowed to freely run the programs and check the output of the test cases. Also, each program is annotated and cross validated by three students. The annotation work takes roughly two months.

After annotation, we observe that 5125 out of the 5196 programs fall into 10 major types of logical errors. The remaining 71 programs have very uncommon errors and are thus discarded from the dataset. The 10 types of logical errors are summarized and explained as follows. The distribution of the numbers of the errors is plotted in Figure 2.

1. **Incorrect input variables** - mainly due to misuse of the ‘&’ operator in the `scanf()` function.
2. **No output** - forgetting to write output.
3. **Incorrect output format** - output format not complying with the exercise requirements.
4. **Incorrect initialization** - errors related to incorrect initialization of variables.
5. **Incorrect data types** - mainly due to undesired type conversions.
6. **Incorrect data precision** - mainly due to loss of precision during calculation.

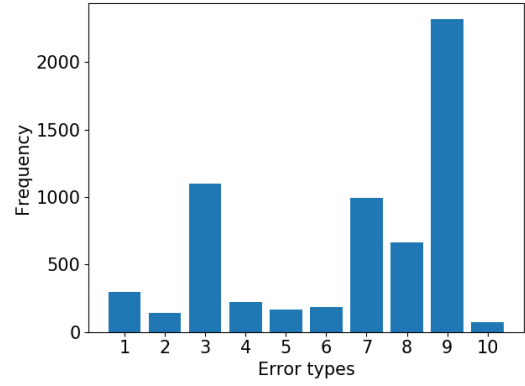


Figure 2: Distribution of the numbers of error types.

7. **Incorrect loops** - loop-related errors such as incorrect termination condition and incorrect step size of iteration.
8. **Incorrect branches** - errors due to incorrect conditional statements.
9. **Incorrect logic** - program’s logic not complying with the exercise.
10. **Incorrect operators** - misuse of operators.

The dataset is randomly splitted into training, validation and testing set with proportion 6 : 2 : 2 in the experiments. We implement PIPE and four baseline models for comparison using Python 3.6 and TensorFlow 1.13. The first three baseline models are the original code2vec model, code2vec plus exercise identity embedding, and code2vec plus exercise identity and evaluation result embeddings. The fourth model is the same as PIPE except that we use the original label-reconstruction loss in the C2AE model. At the input we randomly choose 200 context vectors for each program. At the output all the models are modified to cater for the multi-label classification task accordingly. The batch size is 64 and the learning rate is 0.001. We use the Adam algorithm for optimization. All other optimal hyperparameter settings are determined via the validation process, including the thresholds for rounding to the predicted labels. The metrics of interest are therefore precision, recall and F1 score, in accordance with [2, 24]. We also measure the averaged percentage of exact match, which means that the predicted types of errors are exactly the same as the ground truth for a given program. All experiments are conducted using a normal PC installed with an Intel Core i7-8550U CPU and 8GB RAM.

3.2 Main Results

The main results are presented in Table 1. For PIPE, the program embedding size is set to 138³, the size of the latent space is 69, and the balancing factor $\alpha = 0.1$. For each model, we calculate seven metrics on the testing set, which are the averaged percentage of exact match, per-class precision (C-P), per-class recall (C-R), macro F1 score (Ma-F1), overall precision (O-P), overall recall (O-R) and micro

³program embedding(138)=program vector(64)+exercise ID vector(64)+evaluation result vector(10).

Model	Exact Match	C-P	C-R	Ma-F1	O-P	O-R	Mi-F1
code2vec	0.5735	0.4037	0.3671	0.3822	0.7013	0.6472	0.6714
code2vec + exercise ID	0.5643	0.3696	0.3427	0.3543	0.6978	0.6437	0.6687
code2vec + exercise ID + evaluation	0.5809	0.3798	0.3438	0.3592	0.7088	0.6441	0.6735
PIPE + C2AE loss	0.0	0.09817	0.3679	0.1528	0.2443	0.8497	0.3792
PIPE	0.6259	0.4386	0.3984	0.4151	0.7527	0.7037	0.7255

Table 1: Comparison between PIPE and baseline models.

F1 score (Mi-F1). We observe that PIPE performs constantly much better than the other models on all metrics. Although PIPE using original C2AE loss achieves the best overall recall, it has poor results on all other metrics. This is because the label-reconstruction loss in C2AE attempts to preserve the correlation between the labels and hence more error types are predicted. However, this also drastically reduces the precision and causes no case of exact match is predicted. The results prove the effectiveness of the seamless integration of code2vec and C2AE, as well as the use of cross-entropy for the label-reconstruction loss.

3.3 Sensitivity Analysis

We perform sensitivity analysis for three most important hyperparameters, that is, the size of program embedding v_X , the size of the latent space l and the loss balancing factor α . For each of them, we fix the values of all other hyperparameters and vary it in the corresponding ranges.

The size of program embedding. The size of v_X equals to the sum of the size of program vector v_p , the size of exercise identity vector v_e and the size of evaluation result vector v_r . The size of v_r is fixed to 10 since each exercise has 10 test cases, and the size of v_e is fixed to 64 for the sake of simplicity. We then vary the size of v_p in (64, 128, 192, 256), following the setting in [2]. Therefore, the size of v_X varies in (138, 202, 266, 330). The results are presented in Figure 3. We observe that PIPE prefers smaller program embedding size on all metrics.

The size of the latent space. Following [24], we measure the size of the latent space L as its ratio to the size of program embedding, i.e., $l/|v_X|$. We vary the ratio in the range [0.1, 1] with increments 0.1, and report the results in Figure 4. We observe that roughly all the metrics first increase and then decrease as the size of latent space increases. Overall taking half of the size of program embedding achieves the best performance.

The balancing factor α . We vary α in the range [0.1, 1] with increments 0.1. We also set $\alpha = 0.05$ to show the performance on the very small value. The results are presented in Figure 5. We observe that $\alpha = 0.1$ achieves the best overall performance. Further increasing α would break the balance between the losses of the two parts.

3.4 Demonstration

We have implemented the error-feedback feature in our OJ system using PIPE. Figure 6 shows the usage of the feature. In case of an incorrect submission, a student may check the possible errors predicted by the system and modify the program accordingly, where each type of error is associated with a probability. For example in Figure 6, the student may have

99.04% chance to write incorrect loops, and may also have 93.21% chance to lose precision during calculation, etc.

4. RELATED WORK

Code error prediction (or detection) is a branch of automatic software repair (ASR) [13], which is a long and active research area of software engineering. ASR is with respect to an oracle that is able to determine whether the execution of a given program is correct. Among various types of oracles, test suites or test cases are mostly used in recent ASR researches, which are also used as an important input feature in our PIPE model. Traditionally, test-suite-based methods can be broadly classified into two categories, i.e., search-based methodology [9, 8, 11] and semantics-based methodology [14, 5, 12]. The former category of methods explore a search space of programs to find the most suitable repair candidate that can pass the test cases; the latter category of methods synthesize a repair candidate using semantic information via symbolic execution and constraint solving. All these algorithms are specifically designed for repairing software with thousands to hundreds of thousands lines of code, and often cannot be directly applied in the setting of programming education. For instance, search-based methodologies typically rely on redundancy presented in other parts of the program to limit the search space, whereas redundant code is hardly observed in a students' program. Moreover, rather than directly correcting the bugs in students' programs, providing hints for students to find the errors is more preferable for education purpose. Therefore, the methods for ASR are somehow too heavy to cater for our prediction requirements.

In the past few years, research at the intersection of deep learning and programming languages has been driven by the availability of "big code". Massive source code obtained from the sites such as GitHub as well as some MOOC courses facilitates the design of learnable probabilistic models that exploit abundant patterns of code. These models are then applied to various applications, including program repair [1, 21], clone detection [10] and code synthesis [18], etc.

Training deep learning models to provide feedback to student code has recently drawn attention of both researchers and programming educators. For example, the work of [4] trains recurrent neural networks to automatically detect and correct syntax errors in programming assignments. The models are first trained on syntactically correct student programs and then are used to predict the correct token sequences given the prefix token sequence of a student program with syntax errors. Similarly, the work of [6] trains a multi-layered sequence-to-sequence neural network with attention to predict erroneous locations in student programs and attempts to fix the errors with correct statements. The

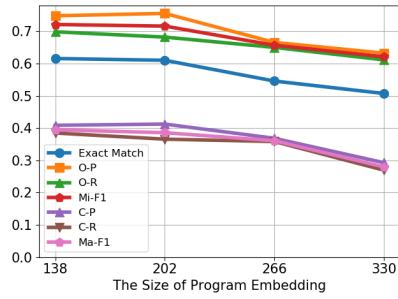


Figure 3: Varying the size of program embedding.

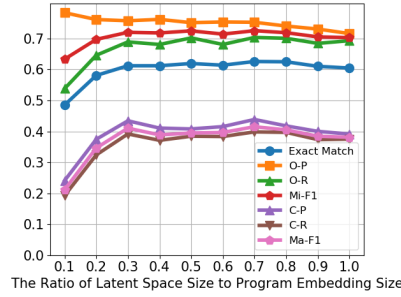


Figure 4: Varying the ratio of the size of latent space to the size of program embedding.

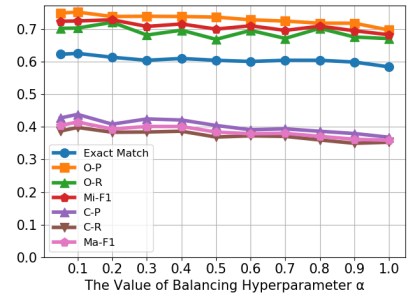


Figure 5: Varying the balancing factor α .

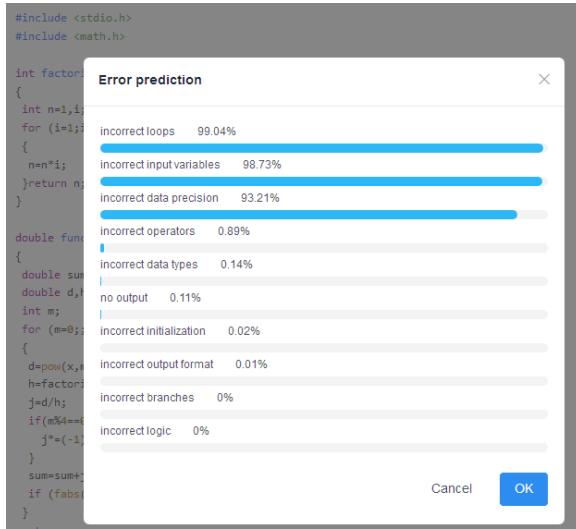


Figure 6: The usage of the error-feedback feature.

model requires to construct training pairs of syntactically incorrect program and the corresponding syntactically correct program. Since both works are focused on detecting and fixing syntax errors, they cannot generate abstract syntax trees for program embedding and thus directly use the language tokens in the original program text. The work in [16] trains an autoencoder to learn joint embedding of program states and programs. The embedding are then used as the input to train an RNN-based model, which can automatically propagate teacher feedback to similar programs. While the focus of their work is representation learning of program state, our model allows end-to-end learning and prediction of logical errors in programs. Other work pertaining to program feedback in the educational setting include [15, 17, 23].

5. CONCLUSIONS

To automate the feedback on logical programming errors in OJ systems, we develop PIPE, a deep learning model that is able to predict the types of errors in students' programs. PIPE seamlessly integrates program representation learning into a multi-label classification model, and thereby can perform end-to-end learning and prediction. To boost the prediction performance, PIPE also incorporates the exercise identity and the evaluation results on the test cases into the

program representation, with the hope that the error information w.r.t each particular exercise and each particular evaluation pattern could be captured. Experimental results on a real dataset show PIPE's superior performance over the baseline models. We have used PIPE to implement the error-feedback feature in our OJ system, and will further evaluate its impact on programming education.

In future, we plan to improve PIPE so that it may not only predict but also localize the errors, i.e., telling the students which lines of the program may contain logical errors and what are the potential types of the errors. Such feedback would further promote students' learning efficiency and help us to achieve higher scalability in programming education.

Acknowledgement

This work was partially supported by the grant from the National Natural Science Foundation of China (Grant No. U1811264).

6. REFERENCES

- [1] M. Allamanis, M. Brockschmidt, and M. Khademi. Learning to represent programs with graphs. In *Proceedings of the International Conference on Learning Representations*, 2018.
- [2] U. Alon, M. Zilberstein, O. Levy, and E. Yahav. code2vec: Learning distributed representations of code. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–29, 2019.
- [3] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, 2013.
- [4] S. Bhatia and R. Singh. Automated correction for syntax errors in programming assignments using recurrent neural networks. In *International Conference on Software Engineering*, 2018.
- [5] F. DeMarco, J. Xuan, D. Le Berre, and M. Monperrus. Automatic repair of buggy if conditions and missing preconditions with smt. In *Proceedings of the 6th international workshop on constraints in software testing, verification, and analysis*, pages 30–39, 2014.
- [6] R. Gupta, S. Pal, A. Kanade, and S. Shevade. Deepfix: Fixing common c language errors by deep learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

- [7] J. R. Kettenring. Canonical analysis of several sets of variables. *Biometrika*, 58(3):433–451, 1971.
- [8] D. Kim, J. Nam, J. Song, and S. Kim. Automatic patch generation learned from human-written patches. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 802–811. IEEE, 2013.
- [9] C. Le Goues, T. Nguyen, S. Forrest, and W. Weimer. Genprog: A generic method for automatic software repair. *Ieee transactions on software engineering*, 38(1):54–72, 2011.
- [10] L. Li, H. Feng, W. Zhuang, N. Meng, and B. Ryder. Cclearner: A deep learning-based clone detection approach. In *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 249–260. IEEE, 2017.
- [11] F. Long and M. Rinard. Staged program repair with condition synthesis. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pages 166–178, 2015.
- [12] S. Mechtaev, J. Yi, and A. Roychoudhury. Directfix: Looking for simple program repairs. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 1, pages 448–458. IEEE, 2015.
- [13] M. Monperrus. Automatic software repair: a bibliography. *ACM Computing Surveys (CSUR)*, 51(1):1–24, 2018.
- [14] H. D. T. Nguyen, D. Qi, A. Roychoudhury, and S. Chandra. Semfix: Program repair via semantic analysis. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 772–781. IEEE, 2013.
- [15] E. Parisotto, A.-r. Mohamed, R. Singh, L. Li, D. Zhou, and P. Kohli. Neuro-symbolic program synthesis. In *Proceedings of the International Conference on Learning Representations*, 2017.
- [16] C. Piech, J. Huang, A. Nguyen, M. Phulsuksombati, M. Sahami, and L. Guibas. Learning program embeddings to propagate feedback on student code. In *Proceedings of the International Conference on Machine Learning*, 2015.
- [17] Y. Pu, K. Narasimhan, A. Solar-Lezama, and R. Barzilay. sk_p: a neural program corrector for moocs. In *Companion Proceedings of the 2016 ACM SIGPLAN International Conference on Systems, Programming, Languages and Applications: Software for Humanity*, pages 39–40, 2016.
- [18] M. Rabinovich, M. Stern, and D. Klein. Abstract syntax networks for code generation and semantic parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2017.
- [19] V. Raychev, M. Vechev, and A. Krause. Predicting program properties from” big code”. *ACM SIGPLAN Notices*, 50(1):111–124, 2015.
- [20] R. T. Rockafellar. Lagrange multipliers and optimality. *SIAM review*, 35(2):183–238, 1993.
- [21] M. Vasic, A. Kanade, P. Maniatis, D. Bieber, and R. Singh. Neural program repair by jointly learning to localize and repair. In *Proceedings of the International Conference on Learning Representations*, 2019.
- [22] S. Wasik, M. Antczak, J. Badura, A. Laskowski, and T. Sternal. A survey on online judge systems and their applications. *ACM Computing Surveys (CSUR)*, 51(1):1–34, 2018.
- [23] M. Wu, M. Mosse, N. Goodman, and C. Piech. Zero shot learning for code education: Rubric sampling with deep learning inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 782–790, 2019.
- [24] C.-K. Yeh, W.-C. Wu, W.-J. Ko, and Y.-C. F. Wang. Learning deep latent space for multi-label classification. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Incidence of teacher curricular emphasis in reading achievement of uruguayan ninth-grade students

Elisa Borba
Instituto Nacional de
Evaluación Educativa
Montevideo, Uruguay
mborba@ineed.edu.uy

Cecilia Emery
Instituto Nacional de
Evaluación Educativa
Montevideo, Uruguay
cemery@ineed.edu.uy

Eliana Lucián
Instituto Nacional de
Evaluación Educativa
Facultad de Humanidades y
Ciencias de la Educación,
Universidad de la República
Montevideo, Uruguay
elucian@ineed.edu.uy

Inés Méndez
Instituto Nacional de
Evaluación Educativa
Montevideo, Uruguay
imendez@ineed.edu.uy

Leonardo Moreno
Facultad de Economía y
Administración, Universidad
de la República
Montevideo, Uruguay
mrleo@iesta.edu.uy

Matías Núñez
Comisión Sectorial de
Enseñanza, Universidad de la
República
Instituto Nacional de
Evaluación Educativa
Montevideo, Uruguay
matias.nunez@cse.udelar.edu.uy

ABSTRACT

Assessing Opportunities to Learn (OTL) implies the measurement of different aspects of the curricular implementation in the classrooms, such as the contents that the teacher selects for the course and the time of exposure and the frequency of the tasks proposed.

Based on recent studies that demonstrate the influence of this curricular dimension on students' performance, this work analyzes the relationships between the emphasis made by teachers of third grade of secondary school on different reading contents, and students' performance in reading tests. The aim of this research is to establish the effect of the emphasis with which teachers propose different types of reading activities (literal, inferential, and critical) on students' performance. From the analysis of compositional data, this study concludes that the students of those teachers who report working a greater extent on the critical dimension of reading obtains –on average– higher scores on the national reading test. This result holds even when socio-economic and cultural context and reading habits are controlled for.

Keywords

Compositional data, curriculum emphasis, opportunity to learn, reading performance, regression models.

1. INTRODUCTION AND BACKGROUND

Leonardo Moreno, Matias Núñez, Cecilia Emery, Inés Méndez, Elisa Borba and Eliana Lucián "Incidence of teacher curricular emphasis in reading achievement of Uruguayan ninth-grade students" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 480 - 486

The opportunities to learn (OTL) that the education systems offer to students are currently one of the central objects of educational evaluations. This implies, among other things, the measurement of various aspects of the curricular implementation in the classrooms, such as: a) the contents that the teacher selects to address in the course and b) their exposure time and frequency according to the tasks proposed. As such, OTL studies have the potential to contribute to the knowledge of what and in which way the curriculum is implemented in the classroom, and to curricular formulations.

One of the first organizations to use the concept of OTL in their studies was the International Association for the Evaluation of Educational Achievement (IEA) in the Second International Study in Mathematics (SIMS), showing the correlation between student performance and the opportunity to have tackled a certain curricular content in class, see [30]. Likewise, the research produced from the SIMS data showed –among other aspects– the centrality of curricular coverage, emphasis, and time devoted to its treatment in relation to students results in tests, see [7].

The OTL project that served as a source for the design of the instruments used in the International Study of Trends in Mathematics and Science (TIMSS), was carried out by the Mathematics and Science Opportunities Survey (MSO) (see [32]), which formulates the concepts of prescribed curriculum (national goals or curricula), implemented curriculum and achieved curriculum (what students have actually learned, see [33]). In this sense, the evaluation of opportunities to learn proposes relevant information about the implemented curriculum that contributes to a better and more complete understanding of what students can learn.

Studies carried out by Cervini conclude that OTL affect stu-

dent performance even when controlling for variables such as socioeconomic status and socio-cultural context of the school: “students with the same social background who attend schools with a similar social composition, but some [have] a significantly higher performance level because their teachers gave them a greater opportunity to learn”, see [25].

Recently, and from the accumulation of evidence, greater importance has been given to the incidence of learning opportunities in student performances. Numerous studies realize the importance of considering these variables when identifying the aspects that influence student performance and that are also modifiable from the organization of schools and the pedagogical practices of the classroom, see [13].

Consequently, when evaluating the performance of students based on tests aligned to the prescribed curriculum (the regulations established to be taught in a school year) it is expected that different results will be observed for those who have had greater coverage of the contents or greater emphasis on its treatment, in addition to more exposure time to them (emphasis), than for those who have not had it, see [25]. The work that is developed in this article aims to give statistical evidence to this last hypothesis raised by Cervini in the Uruguayan classrooms.

On a large scale, there are few international experiences that evaluate learning opportunities in the classroom: the evaluations carried out by the IEA (SMSO and TIMSS), the International Study of Progress in Reading Comprehension (PIRLS), the Program for the International Assessment of Students (PISA) and the International Survey on Teaching and Learning (TALIS). Since 2017, the National Institute of Educational Evaluation (INEEd) of Uruguay, performs the national evaluation of achievements of the educational system (ARISTAS) that contemplates the measurement of OTL in classrooms of third and sixth grade of primary education and third grade of secondary education.

Research based on PISA data (see [17]) has shown, for a large group of countries, a significant association (of moderate to strong) between reading habits (students’ taste for reading, time dedicated to it in their free time and the diversity of texts they read) and their reading performance.

Similar findings were reported for Uruguay by ANEP, 2004. In this study, the relationship between certain attitudinal variables regarding students’ reading habits and the results of reading and math tests is analyzed. In concordance with international findings, this study shows better reading results for women than men. Nonetheless, it is also women who show greater taste for reading. Among the findings of this study, it is noted that as performance in reading improves, so does the taste for reading, while the dislike of it decreases on average.

Although there is evidence that links student reading habits with their reading performance. This can happen since student reading habits allow some teacher practices, that in turn influence student performance. However, there is little evidence that links these qualities of students with teacher practices (see [4]) and their performance. Thus, this research focuses on the relative effect of teaching practices on

the reading performance of students, when certain student qualities (such as socioeconomic and cultural level; and student reading habits) are controlled for.

Based on data from the 2018 national evaluation of the Uruguayan educational system –Aristas– carried out by the National Institute of Educational Evaluation (INEEd), the aim of this work (considering recent studies that demonstrate the influence of curricular implementation on student performance) is to analyze the relationships between the emphasis made by nine grade teachers on different reading dimensions (literal, inferential and critical) and their student performance on reading.

One possible explanation is that the students whose teachers report working the critical dimension of reading on a greater extent, achieve –on average– higher scores in the reading test, even when socio-economic and cultural context; and reading habits, are controlled for.

This paper is organized as follows. In Section 2 the theoretical framework and the main linguistic concepts used are introduced. Section 3 provides the implemented methodology and the statistical tools used: Compositional Data Analysis (CoDA). The most relevant results obtained from the statistical analysis are enumerated in Section 4. In Section 5 the final discussion and conclusions are established.

2. THEORY

As Zakaryan (see [35]) indicates, in general terms, the nature of classroom teaching significantly affects the level of student learning. In this relationship, the analysis of the teaching practice is key, since the teacher determines the learning opportunities through the activities he proposes in the classroom and the qualities of his or her teaching. Thus, the content, format and cognitive demand of the tasks that teachers pose in the classroom constitute “the main vehicle to provide school children with learning opportunities”, see [28], p. 113.

One of the questions which OTL studies attempt to respond is whether the curriculum implemented by teachers in the classrooms effectively covers the contents established in the nationally prescribed curriculum. Within this research perspective, three approaches are identified: the coverage of the contents, the exposure to them –measured through the time dedicated to them– and the emphasis on their implementation, that is, what contents are treated with priority in the classrooms over others in the program, see [26].

In this way, it is possible to approach the institutional and pedagogical mechanisms that contribute to the distribution of learning opportunities not only among schools, but also between classrooms. Likewise, it is relevant to know the form of said distribution, that is, if all children and young people receive from the school system –or not– the same opportunities to learn.

This work follows the definition of opportunities to learn (OTL) carried out by the INEEEd, see [15]. In this, the study of the OTL not only seeks to analyze the alignment between the prescribed and the implemented curriculum, but also to what extent this alignment and other school conditions in-

fluence the performance of the students. Among the dimensions of the OTL assessed by the INEE, this paper focuses on coverage and sequence of the contents, which refers to the degree of implementation of the curricular contents, as well as their didactic sequence and emphasis, see [15], p. 14. In particular, the emphasis on curricular content is measure as the percentage of classroom pedagogical time that teachers devote to the work of curricular content corresponding to a school grade, see [15], p. 15.

3. METHOD

This study uses data from the evaluation of the uruguayan education system (Aristas) carried out by the INEE in secondary education in 2018 nationwide. This evaluation gathers information on various actors of the education system: school principals, teachers and students, through self-administered online surveys. Aristas is applied to a representative sample of nine grade middle school students of the Uruguayan urban region. Besides this contextual questionnaires, each student has to take two multiple choice tests to assess reading and math performance (Students' abilities are estimated based on test results using item response theory, and in particular the Rash model).

To obtain information on curricular implementation, Aristas asks teachers if they have covered and with how much emphasis, a series of curricular activities that can be classified into three reading dimensions: literal, inferential and critical.

As defined by INEE in the OTL framework for secondary education (see [14] and [15]), the emphasis with which teachers work the curricular contents in the classroom is a composite measure based on their reports on: i) the total number of classes taught and ii) the number of classes dedicated to addressing each type of activity in the course. Both measures refer to the period between the beginning of the school year (March 2018) and the time the evaluation was carried out (October 2018).

In order to address this, teachers are presented with a list of 10 activities (corresponding to the three different dimensions of reading –literal, inferential, critical–) and are asked to indicate the number of classes dedicated to each activity. Each of the dimensions in which the activities are grouped implies differences in their cognitive complexity, see Table 1. Thus, the number of classes reported per dimension is counted as the average number of classes declared for each of the listed activities. That is, the number of classes reported by a teacher in one dimension is the sum of classes that the teacher declares to dictate for the activities that integrate that dimension, over the total of activities that compose it.

From the self-report declared by nine grade teachers in secondary education, an overestimation of the effective number of classes in the school year is generally observed. That is, when the maximum total number of classes in the year is 72%, 70.2% report dictating above that threshold. This inconvenience in the instrument is a frequent problem due to various causes, see [10] and [4].

To approach this over report problem, the classes reported are relativized by each teacher for the activities of each di-

	Literal dimension	Inferential dimension	Critical dimension
Kinds of activity	Recognize basic elements of the enunciation situation	Recognize the subject of the paragraph or statement	Evaluate and / or interpret the facts, situations or concepts posed by the text
	Locate explicit information	Summarize the general idea of the text and draw conclusions	Recognize items complexes of the enunciation situation (assumptions, implications, reasons, ideological position of the enunciator, intertextuality, parody, irony, exaggeration)
	Recognize the thematic progression	Recognize narrative, descriptive, argumentative or expository intentionality	
		Match information of sentences and paragraphs	
		Hierarchize data or events and establish relationships between texts when it has different formats (eg tables and text)	

Table 1: Activities classified according to the dimension they conform. Source: own elaboration.

mension of reading to the total of classes according to the school calendar. It is assumed that the relative weight that the teacher reports in each activity is a valid measure of the real time spent in class.

The classes reported by the i -th teacher to the literal, inferential and critical reading respectively are called L_i , I_i and C_i , so the normalized vector is

$$X_i = (LR_i, IR_i, CR_i) = \frac{1}{L_i + I_i + C_i} (L_i, I_i, C_i).$$

Let S^2 be the unit simplex of \mathbb{R}^3 ($S^2 = \{(x, y, z) \in \mathbb{R}^3 / x + y + z = 1, x \geq 0, y \geq 0, z \geq 0\}$). It is true that $X_i \in S^2 \forall i = 1, \dots, N$, where N is the total number of teachers in the sample ($N = 364$). In this way, conclusions can be drawn about the behavior in relative but not absolute terms. The isometric transformation is then given by $\phi: R_{\geq 0}^3 \rightarrow S^2$, where $\phi(x, y, z) = \left(\frac{x}{x+y+z}, \frac{y}{x+y+z}, \frac{z}{x+y+z} \right)$ y $R_{\geq 0}^3 = \{(x, y, z) \in \mathbb{R}^3 / x, y, z \geq 0\}$.

This means that the class proportions dictated (on average) in each of the three dimensions of reading (literal, inferential and critical) are thought of as compositional data. In this sense, the vector is made up of three components, each of which represents the proportion of classes dictated in that dimension, therefore the components of the vector of proportions are non-negative and add to 1. This vector indicates the relative emphasis the teacher reports assigning to each of the dimensions. This type of compositional data analysis doesn't allow for classical or hierarchical regression models as its assumptions don't hold.

The analysis of compositional data dates back to Pearson 1987, but the basis of statistical theory for this type of data has been developing since the middle of the last century, see [12], [8] and [9]. The association that exists between the components of the vector determines a series of methodological difficulties (see [34]) that lead to the need for specific techniques for this type of data.

A large number of applications of compositional data theory are found in the literature for different fields (see for example [5], [23], [20] and [27]). However, there are few applications in reference to educational data and in particular in the area

of educational evaluation, see for example [31], [19], [21] and [6].

On the other hand, reading achievement is assessed through a standardized reading test developed by INEEd. The test is designed based on a general statement of reading (reading competence) which is successively broken down into statements (dimensions of said competence: literal, inferential and critical) and subaffirmations (knowledge and skills). Test items are designed to assess students mastery of the subaffirmations. From a psychometric perspective, the Item Theory Response (TRI) method is used for the calibration and construction of the items. For the porpoise of this study, the group performance or ability (Y_i) is the average of the scores assigned by TRI to all the students of the i th-group.

4. RESULTS

Data from Aristas 2018 in Uruguay shows a relationship between socioeconomic level and reading habits of the group, and the emphasis the teacher assigns to certain reading activities over others, see [4]. Higher socioeconomic levels and more frequent reading habits of the group are associated with more teacher emphasis on the dimension of critical reading. Even when socioeconomic level is controlled for, the reading habits of the group have a significant effect over teacher emphasis.

Figure 1 shows, by means of box plot, the strong relationship between the quintiles of the socioeconomic level of the group (ESCS) and the average reading performance obtained by the group of students (left panel). Moreover, higher reading habits of the group also have an effect, (not as pronounced as ESCS) on student performance (right panel).

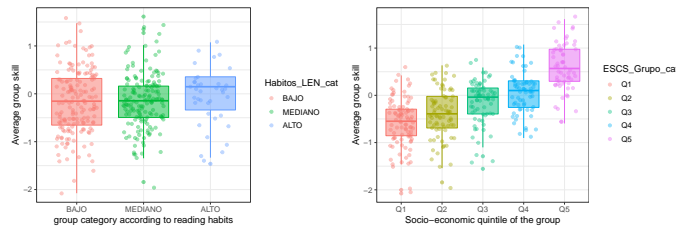


Figure 1: Left Panel: Box plot of the average performance of the group in the reading test according to the socioeconomic quintile of the group. Right Panel: Box plot of the average performance of the group in the reading test according to the level of reading habits of the group.

The relationship between reading test results and emphasis placed by teachers on each of the three dimensions of reading, is evaluated. As Figure 2 shows, groups who perform better on the test devote more time to activities that involve critical reading. Moreover, as group performance improve, the proportion of classes dedicated to literal reading decreases. These trends don't appear to apply to inferential reading.

However, these results are a) only descriptive of the sample and b) they are marginal, that is, the joint effect of the activity vector on student performance and its relative importance vis-a-vis other variables of the study (ESCS and

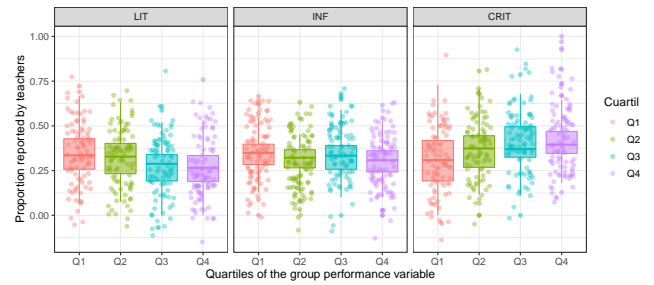


Figure 2: Box diagrams of the proportion in each of the activities (literal-inferential-critical) according to the quartiles of the group's performance variable.

reading habits) is not observed.

To model the problem, a non-parametric regression model (kernel method, see [18]) and a parametric regression model (Dirichlet regression, see [29] and [24]) are considered, see Figure 3. This figure reflects the average percentages of emphasis placed on each dimension as a function of the group's performance. On average, according to the non-parametric model, the emphasis on literal reading is among 20% and 35%, decreasing as Y increases. The influence of Y on the proportion of activities in critical reading varies between an average range of 27% to 50% increasing as Y increases. The inferential reading is the one with the least variation as a function of Y , between 30% and 38%.

In both cases it is observed that increasing group's performance is associated to an increase in the proportion of classes in which critical reading activities are taught.

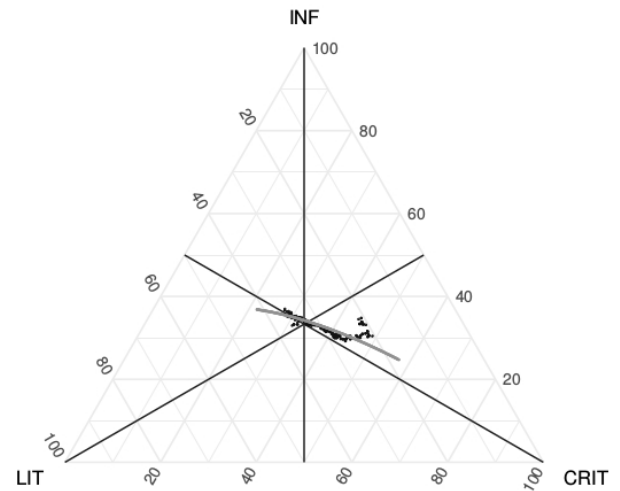


Figure 3: Non-parametric regression by kernel method (points) and Dirichlet regression (continuous line) in S^2 where the independent variable is the performance of the group (Y).

This result holds for the Dirichlet model. In this case the decrease of literal reading emphasis is associated with higher test results but with a moderate statistical significance (p -value = 0.07), see Table 2.

Coef.	ANOVA		
Lit.	Coef.	Error	p-value
Int.	0.86	0.06	< 2e-16
Y	-0.14	0.08	0.07
Inf.	Coef.	Error	p-value
Int.	0.97	0.06	< 2e-16
Y	0	0.07	0.99
Crit.	Coef.	Error	p-value
Int.	1.12	0.06	< 2e-16
Y	0.46	0.07	8.94e-11

Table 2: ANOVA of the Dirichlet regression where the independent variable is the performance of the group (Y).

A possible mistake when interpreting these results is to infer a direct association between emphasis and student performance when, strictly speaking, it could be an indirect effect caused by student reading habits and socio-economic and cultural level of the group, see [4]. Figure 4 shows these interactions.

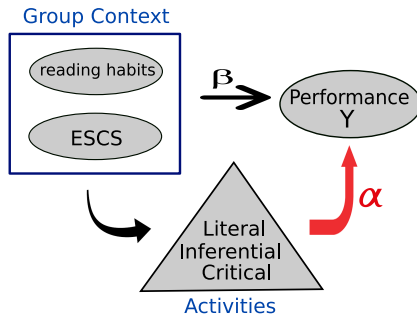


Figure 4: Diagram on the relationship between activities and performance controlling for ESCS context variables and study habits.

To control the effect of socioeconomic status and reading habits on performance, two step regression is estimated. The performance of the group in reading Y can be decomposed as the sum of a β (hypothetical or latent) score generated by certain context variables (in this model they are the *ESCS* and reading habits noted as H) and another α associated with the emphasis of classes in each activity, that is

$Y = \text{Intercept} + \beta + \alpha + \epsilon$, where ϵ is a random error centered and independent of α and β . In this case, student performance is considered as the dependent variable (as opposed as in Figure 3).

The strong relationship between β and Y has already been widely studied, see [14]. If the last term α is also significant to explain performance, this indicates that emphasis on a certain type of activities also influences performance, even controlling for certain group context variables. To “extract” the effect of the context, the following procedure is performed. As a first step, to avoid the ipstative effect (see [11]) between the components of the vector in S^2 an isometric transformation is carried out that goes from S^2 to \mathbb{R}^2 . This is called the log-ratio isometric transformation (*ilr*), see [1]. The image of (LR, IR, CR) is noted by (ilr_x, ilr_y) .

The first adjustment is then a linear model of the form,

$$Y = a_0 + a_1 ilr_x + a_2 ilr_y + b_1 ESCS + b_2 H + \epsilon \quad (\text{Model 1}).$$

Table 3 shows how all the variables are significant in this model, with the variables of S^2 tolerating the inclusion of H and *ESCS* in the model.

Coef.	ANOVA		
Model 1	Coef.	Error	p-value
Intercept	-0.11	0.025	4.06e-06
<i>ilr_x</i>	0.15	0.050	0.01820
<i>ilr_y</i>	0.12	0.042	0.00586
<i>ESCS</i>	0.68	0.04	< 2e-16
<i>H</i>	0.32	0.082	0.00013
Model 2	Coef.	Error r	p-value
Intercept	-0.10	0.026	< 3.44e-05
<i>ilr_x</i>	0.15	0.053	0.00465
<i>ilr_y</i>	0.14	0.044	0.00146

Table 3: ANOVA considering *ESCS* and *H* as control variables.

Second, after estimating the coefficients of the equation for maximum likelihood, the effect on the performance of the context variables is extracted, that is, $Y_i^* = Y_i - \hat{b}_1 ESCS_i + \hat{b}_2 Habitos_i$. The final model (model 2) that allows us to deduce the effect of emphasis on each dimension is,

$$Y^* = d_0 + d_1 ilr_x + d_2 ilr_y + \epsilon^* \quad (\text{Model 2}).$$

The ANOVA results of the Model 2 are also found in Table 3.

If the inverse *ilr* transformation is performed to the estimated coefficient vector of Model 2, the coefficient vector in S^2 is obtained, $(LIT, INF, CRIT) = (0.28, 0.35, 0.37)$ which allows to conclude that the emphasis on certain activities allows to obtain a better performance, still subtracting the effect of the context.

Figure 5 shows the level curves of the estimated regression function in Model 2. Even when controlling for the context variables, teachers who emphasize the activities of critical reading versus literal reading obtain –on average– a better group performance in the reading test. Higher intensities of gray indicate larger group performance.

It is important to highlight that Model 2 has high variability, which makes it possible to conjecture that other group, school or teacher variables that influence performance are not being accounted for in the model.

5. DISCUSSION AND CONCLUSIONS

This paper analyzes the relationship between the emphasis placed by teachers on different classroom reading activities (literal, inferential and critical reading), and the reading performance of their nine grade students in Uruguay. This makes it a novel contribution to educational policy in Uruguay as the topic has never been studied, and it is also a contribution to educational evaluation, as it applies the analysis of compositional data to this field of research.

Findings show a relationship between reading test scores and the emphasis placed by teachers on different reading

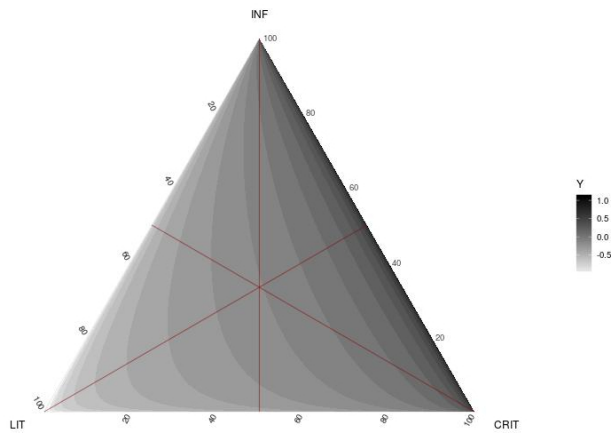


Figure 5: Level curves set of the estimated regression function in Model2.

dimensions. Better test results are associated with teacher emphasis on critical reading activities. Nonetheless, this relationship is not direct as it is affected by other variables such as the socioeconomic and cultural level of the group of students and their reading habits. However, even when controlling for these variables, the evidence holds: greater emphasis placed on critical reading activities versus literal reading is associated with higher group results on the reading test.

Moreover, findings also suggest the omission of certain variables that can be relevant to explain reading performance by students. It could be the case of prior reading achievement. Although it could be argued that teachers place more emphasis on critical reading activities when their students are better prepared for that (when they have higher prior reading achievement), evidence from primary school students from Uruguay shows that is not necessarily the case, see [16]. Nonetheless, this will be explored in future research.

In Uruguay, the differences in academic achievement are strongly related to socioeconomic factor, see [23], [2] and [3]. This study shows that having those factors controlled, the OTL offered to student variable is still relevant on achievement data. The dimension of this finding has implications not only on the teaching profession but also on public educational policies.

6. REFERENCES

- [1] J. Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):139–160, 1982.
- [2] ANEP. Uruguay en pisa 2012. primer informe. Montevideo, Uruguay, 2014.
- [3] ANEP. Evaluación nacional de 6to año. en matemática, ciencias y lengua. 2013. primer informe. Montevideo, Uruguay, 2015.
- [4] E. Borba, C. Emery, I. Mendez, E. Lucian, L. Moreno, and M. N. nez. Incidence of students' home reading habits in teachers' classroom proposals in middle education in uruguay (in review).
- [5] A. Buccianti, G. Mateu-Figueras, and V. Pawlowsky-Glahn. Compositional data analysis in the geosciences: From theory to practice. Geological Society of London, 2006.
- [6] R. D. Burns, Y. Kim, W. Byun, and T. A. Brusseau. Associations of school day sedentary behavior and physical activity with gross motor skills: use of compositional data analysis. *Journal of Physical Activity and Health*, 16(10):811–817, 2019.
- [7] L. Burstein. Completed, ongoing, and projected. *Second International Mathematics Study: Studies*, page 25, 1990.
- [8] J. C. Butler. Visual bias in r-mode dendrograms due to the effect of closure. *Mathematical Geology*, 10(2):243–252, 1978.
- [9] J. C. Butler. The effects of closure on the moments of a distribution. *Journal of the International Association for Mathematical Geology*, 11(1):75–84, 1979.
- [10] V. Cabezas, M. P. Medeiros, D. Inostroza, C. Gómez, and V. Loyola. Organización del tiempo docente y su relación con la satisfacción laboral: Evidencia para el caso chileno. *Education Policy Analysis Archives/Archivos Analíticos de Políticas Educativas*, (25):1–33, 2017.
- [11] W. Chan and P. M. Bentler. The covariance structure analysis of ipsative data. *Sociological Methods & Research*, 22(2):214–247, 1993.
- [12] F. Chayes. On correlation between variables of constant sum. *Journal of Geophysical research*, 65(12):4185–4193, 1960.
- [13] S. Cueto, J. León, C. Ramírez, and G. Guerrero. Oportunidades de aprendizaje y rendimiento escolar en matemática y lenguaje: resumen de tres estudios en Perú. *REICE: Revista Iberoamericana sobre Calidad, Eficacia y Cambio en Educación*, 6(1):29–41, 2008.
- [14] I. N. de Estadística. Informe sobre el estado de la educación en Uruguay 2015-2016, 2017.
- [15] I. N. de Evaluación Educativa-INEEd. Marco de oportunidades de aprendizaje en tercero de educación media, 2018.
- [16] I. N. de Evaluación Educativa-INEEd. Reporte de aristas 1. las oportunidades de aprendizaje en Uruguay: diagnóstico y tratamiento de contenidos curriculares en las aulas de primaria, 2019.
- [17] P.-K. Deutsches et al. Pisa 2000. *Basiskompetenzen von Schülerinnen und Schülern im internationalen Vergleich*. Opladen, 29, 2001.
- [18] M. Di Marzio, A. Panzera, and C. Venieri. Non-parametric regression for compositional data. *Statistical Modelling*, 15(2):113–133, 2015.
- [19] D. Dumuid, T. Olds, J.-A. Martín-Fernández, L. K. Lewis, L. Cassidy, and C. Maher. Academic performance and lifestyle behaviors in Australian school children: a cluster analysis. *Health Education & Behavior*, 44(6):918–927, 2017.
- [20] D. Dumuid, T. E. Stanford, J.-A. Martin-Fernández, Ž. Pedišić, C. A. Maher, L. K. Lewis, K. Hron, P. T. Katzmarzyk, J.-P. Chaput, M. Fogelholm, et al. Compositional data analysis for physical activity, sedentary time and sleep research. *Statistical methods in medical research*, 27(12):3726–3738, 2018.
- [21] D. Dumuid, T. E. Stanford, Ž. Pedišić, C. Maher,

- L. K. Lewis, J.-A. Martín-Fernández, P. T. Katzmarzyk, J.-P. Chaput, M. Fogelholm, M. Standage, et al. Adiposity and the isotemporal substitution of physical activity, sedentary time and sleep among school-aged children: A compositional data analysis approach. *BMC Public Health*, 18(1):311, 2018.
- [22] T. Fernández and S. Cardozo. Tipos de desigualdad educativa, regímenes de bienestar e instituciones en américa latina: un abordaje con base en pisa 2009. *Páginas de educación*, 4(1):33–55, 2011.
- [23] B. Ferrer-Rosell, G. Coenders, and E. Martínez-García. Segmentation by tourist expenditure composition: an approach with compositional data analysis and latent classes. *Tourism analysis*, 21(6):589–602, 2016.
- [24] R. H. Hijazi and R. W. Jernigan. Modelling compositional data using dirichlet regression models. *Journal of Applied Probability & Statistics*, 4(1):77–91, 2009.
- [25] R. C. Iturre. Efecto de la” oportunidad de aprender” sobre el logro en matemáticas en la educación básica argentina. *REDIE: Revista Electrónica de Investigación Educativa*, 3(2):1, 2001.
- [26] D. Lafontaine, A. Baye, S. Vieluf, and C. Monseur. Equity in opportunity-to-learn and achievement in reading: A secondary analysis of pisa 2009 data. *Studies in Educational Evaluation*, 47:1–11, 2015.
- [27] M. L. C. Leite. Applying compositional data methodology to nutritional epidemiology. *Statistical methods in medical research*, 25(6):3057–3065, 2016.
- [28] J. L. Lupiáñez. *Expectativas de aprendizaje y planificación curricular en un programa de formación inicial de profesores de matemáticas de secundaria*. PhD thesis, Universidad de Granada, 2010.
- [29] M. J. Maier. Dirichletreg: Dirichlet regression for compositional data in r. 2014.
- [30] B. O. Muthén, C.-F. Kao, and L. Burstein. Instructionally sensitive psychometrics: Application of a new irt-based detection technique to mathematics achievement test items. *Journal of Educational Measurement*, 28(1):1–22, 1991.
- [31] K. Namboodiri, R. G. Corwin, and L. E. Dorsten. Analyzing distributions in school effects research: An empirical illustration. *Sociology of Education*, pages 278–294, 1993.
- [32] W. H. Schmidt and C. C. McKnight. Surveying educational opportunity in mathematics and science: An international perspective. *Educational evaluation and policy analysis*, 17(3):337–353, 1995.
- [33] W. H. Schmidt, M. T. Tatto, K. Bankov, S. Blömeke, T. Cedillo, L. Cogan, S. I. Han, R. Houang, F. J. Hsieh, L. Paine, et al. The preparation gap: Teacher education for middle school mathematics in six countries. *MT21 Report. East Lansing: Michigan State University*, 32(12):53–85, 2007.
- [34] K. G. Van den Boogaart and R. Tolosana-Delgado. *Analyzing compositional data with R*, volume 122. Springer, 2013.
- [35] D. Zakaryan. El tipo de tareas como oportunidad de aprendizaje y competencias matemáticas de estudiantes de 15 años. 2013.

Moving beyond Test Scores: Analyzing the Effectiveness of a Digital Learning Game through Learning Analytics

Huy Anh Nguyen
Carnegie Mellon University
hn1@cs.cmu.edu

Xinying Hou
Carnegie Mellon University
xhou@cs.cmu.edu

John Stamper
Carnegie Mellon University
jstamper@cs.cmu.edu

Bruce M. McLaren
Carnegie Mellon University
bmclaren@cs.cmu.edu

ABSTRACT

A challenge in digital learning games is assessing students' learning behaviors, which are often intertwined with game behaviors. How do we know whether students have learned enough or needed more practice at the end of their game play? To answer this question, we performed post hoc analyses on a prior study of the game *Decimal Point*, which teaches decimal numbers and decimal operations to middle school students. Using Bayesian Knowledge Tracing, we found that students had the most difficulty with mastering the number line and sorting skills, but also tended to over-practice the skills they had previously mastered. In addition, using students' survey responses and in-game measurements, we identified the best feature sets to predict test scores and self-reported enjoyment. Analyzing these features and their connections with learning outcomes and enjoyment yielded useful insights into areas of improvement for the game. We conclude by highlighting the need for combining traditional test measures with rigorous learning analytics to critically evaluate the effectiveness of learning games.

Keywords

Decimal, Digital Learning Game, Bayesian Knowledge Tracing, Over-practice

1. INTRODUCTION

Digital learning games are typically regarded as a powerful tool to promote learning by engaging students with a novel and interactive game environment. While there have been concerns about the lack of empirical results on learning games' effectiveness [21, 32], recently we have seen more research that addresses this issue by showing students' learning gains from pretest to posttest in rigorous randomized experiments [9, 41, 52]. More generally, a meta-analysis of 69

studies by [10] showed that game conditions promoted significantly more learning than non-game conditions with equivalent knowledge content, and that augmented game designs with more learning-oriented features were more instructionally effective than standard designs.

While this prior research has demonstrated that digital learning games can enhance learning, the next step is to examine how they do so. In particular, even though the common measures of pretest and posttest scores are necessary to evaluate students' transferable learning, they are inadequate to address many questions about how learning takes place during the game. For example, did students get just enough practice from the game, or more practice than necessary? How does in-game learning correlate with test performance? These questions have been explored in great detail in Intelligent Tutoring Systems (ITS), but not as much in digital learning games, primarily because of the differences in design approaches between these two platforms. ITS are typically very structured environments where students are frequently evaluated on their knowledge and, in the mastery learning settings [28], move to a new skill as soon as the system determines they have mastered the current skill. In contrast, digital learning games emphasize students' freedom in shaping their own learning experience without concern about the consequences of failure [15]; as a result, the game's learning objectives are not always obvious to the students [4]. The question, then, is how can we combine the traditional pretest and posttest measures in learning game studies with learning analytics methods from ITS to paint a better picture of students' learning, both inside and outside of the game context? Furthermore, given the game's dual goal of promoting both learning and enjoyment, do in-game learning metrics also relate to students' enjoyment in any meaningful way?

Our work explores these questions in the context of *Decimal Point*, a game that teaches decimal numbers and operations to middle-school students. Here we present a post hoc analysis of the data from a prior study [22]. First, we investigated how well students mastered the in-game skills, how long it took them to master each skill, and whether students continued practicing after mastery. Next, we used student data from before and during game play to predict their learning outcomes and enjoyment after the game. Based on this re-

Huy Nguyen, Xinying Hou, John Stamper and Bruce McLaren "Moving beyond Test Scores: Analyzing the Effectiveness of a Digital Learning Game through Learning Analytics" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 487 - 495

sult, we derived lessons for improving learning support in *Decimal Point* as well as in a more general learning game context.

2. RELATED WORK

2.1 Learning Analytics in Games

In-game formative assessment can be a powerful complementary tool for capturing students' learning progress [59]. Traditional formative measures typically make use of game-based metrics, such as the number of completed levels or the highest level beaten [2, 11], but these metrics may not always align with actual learning. Prior studies on *Decimal Point*, for instance, reported that students who played more mini-game rounds did not learn more than those who played fewer [18, 39]. An alternative approach is to employ learning analytics methods from ITS studies. For example, learning curve analysis, which visualizes students' error rates over time, has been applied in several learning games and yielded valuable insights that range from instructional redesign lessons to discovery of unforeseen strategy by students [17, 29, 42].

Learning analytics techniques can also connect formative assessment with external performance. For example, Bayesian networks have been applied to predict posttest responses from students' in-game data in several learning games [30, 48, 54]. Similarly, [27] employed feature engineering and gradient boosted random forest algorithm to identify struggling students in real-time in a physics learning game. Recently we have also seen more usage of deep learning for this prediction task [24, 51]. In general, research work in this direction can illustrate how well students' learning aligns with the game's learning objectives, while also guiding the development of adaptive support game features.

2.2 Decimal Point

Decimal Point is a web-based single-player digital learning game that helps middle-school students learn about decimal numbers and their operations (e.g., adding and comparing). The game features an amusement park metaphor, with a map of the park used to guide students (Figure 1). There are 8 theme areas with 24 mini-games, connected by a line that is designed to interleave skill types and theme areas. Each mini-game is aimed at helping students solve one of the common decimal misconceptions: **Megz** (longer decimals are larger), **Segz** (shorter decimals are larger), **Pegz** (the two sides of a decimal number are separate and independent) and **Negz** (decimals smaller than 1 are treated as negative numbers) [25]. Also, each mini-game calls for one of the following skills:

1. **Addition**: add two decimals by entering the carry digits and the sum.
2. **Bucket**: compare given decimals to a threshold number and place each decimal in a "less than" or "greater than" bucket.
3. **Number Line**: locate the position of a decimal number on the number line.
4. **Sequence**: fill in the next two numbers of a sequence of decimal numbers.
5. **Sorting**: sort a list of decimal numbers in ascending or descending order.

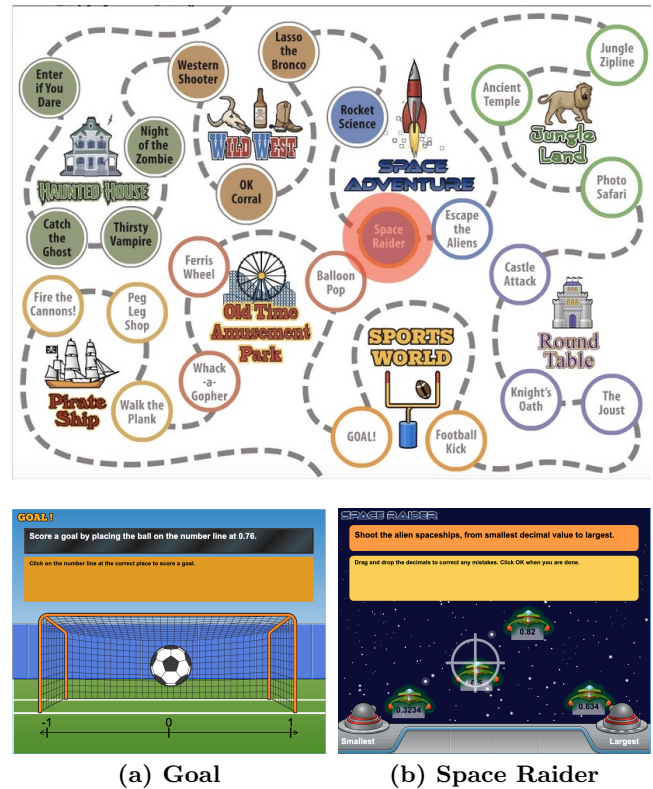


Figure 1: Screenshots of the main map screen and two example mini-games. Goal is a Number Line game and Space Raider is a Sorting game.

In each mini-game, students solve a number of decimal problems related to the game's targeted skill and receive immediate feedback about the correctness of their answers. Students don't face penalty on incorrect responses and can re-submit answers as many times as needed; however, they are not allowed to move forward without solving all the problems in the mini-game. More details about the instructional content of the mini-game problems can be found in [35].

The original study of *Decimal Point* showed that the game led to more learning and enjoyment than a conventional tutor with the same instructional content [35]. Subsequent studies have integrated the element of agency into the game, by endorsing students to select their preferred mini-games to play and stopping time [18, 39]. Based on their findings, students who were provided agency acquired equivalent learning gains in less time than those who were not. Most recently, a study by [22] compared two versions of the game, one that encourages students to play to learn, and one that encourages them to play for fun. Their results indicated that the learning-oriented group focused on re-practicing the same mini-games, while the enjoyment-oriented group did more exploration of different mini-games. In general, while all of these previous works reported that students learned from the game across all study conditions, it is not yet clear which game factors contributed to these findings. Furthermore, no connection between students' learning and their enjoyment has been identified. Our work aims at acquiring more insights into these areas.

Table 1: Survey items before and after game play.

Pre-intervention surveys		
Dimension (item count)	Example statement	Cronbach's α
Decimal efficacy (3) [44]	I can do an excellent job on decimal number math assignments.	.83
Computer efficacy (3) [31]	I know how to find information on a computer.	.71
Identification agency (2) [50]	I work on my classwork because I want to learn new things.	.60
Intrinsic agency (2) [50]	I work on my classwork because I enjoy doing it.	.86
External agency (3) [50]	I work on my classwork so the teacher won't be upset with me.	.61
Perseverance (3) [12]	Setbacks don't discourage me. I don't give up easily.	.79
Math utility (3) [13]	Math is useful in everyday life.	.63
Math interest (2) [14]	I find working on math to be very interesting.	.75
Expectancy (1) [23]	I plan to take the highest level of math available in high school.	-
Post-intervention surveys		
Dimension (item count)	Example statement	
Affective engagement (3) [5]	I felt frustrated or annoyed.	.78
Cognitive engagement (3) [5]	I tried out my ideas to see what would happen.	.54
Game engagement (5) [7]	I lost track of time.	.74
Achievement emotion (6) [43]	Reflecting on my progress in the game made me happy.	.89

3. DATASET

Our work uses data from 159 fifth and sixth grade students in our prior study [22], where students could select and play the mini-games from the map in Figure 1 in any order, and were allowed to stop playing at any time after finishing 24 mini-game rounds. They could also play more rounds of the completed mini-games, with the same game mechanics but different question content. For example, the first round of the mini-game *Goal* asks students to locate 0.76 on the number line, while the second round features the same game interactions but involves locating 0.431. Before playing, students did a pretest and answered demographic survey questions. After game play, they completed another survey to evaluate their experience and did a posttest, followed by a delayed posttest one week later. Here we outline the measures which are relevant to our analyses. A more detailed description of the experimental design can be found in [22].

Pretest, Posttest, and Delayed Posttest: Each test consisted of 43 items, for a total of 52 points. The items were designed to probe for specific decimal misconceptions, and involved either the five decimal skills targeted by the game or conceptual questions (e.g., “is a longer decimal larger than a shorter decimal?”). There are three test versions (A, B and C), which are isomorphic to one another and counterbalanced across students (e.g., ABC, ACB, BAC, etc. for pre, post, and delayed). Our prior analysis showed no differences in difficulty between the three versions [22].

Questionnaires: Before game play, students reported their age and gender, as well as their ratings to survey items about their background information, from 1 (“Strongly Disagree”) to 5 (“Strongly Agree”). After playing, students rated their

enjoyment (also from 1 to 5) via survey questions that address four enjoyment dimensions (Table 1). If a dimension comprises several items, we compute the average ratings of all items in that dimension to derive its representative rating score. According to [16], a measure should have $\alpha \geq .60$ to be considered reliable; therefore, based on Table 1 we removed the cognitive engagement dimension (with $\alpha = .54$) from further analyses.

The full log data from the study is archived in the DataShop repository [55], in dataset number 3086. We present our analysis of this data in the following section.

4. RESULTS

4.1 Investigating in-game learning

In our prior work on Knowledge Component (KC) modeling in *Decimal Point*, based on data from a separate study, we used the correctness of the student's first attempt in answering each mini-game problem to update their mastery of the KC covered by that mini-game. With this mapping from in-game action to KC, we found that students' learning can be better captured by a KC model based on skill types (e.g., *Addition*, *Bucket*) than on decimal misconceptions (e.g., *Segz*, *Negz*) [40]. Therefore, in this work we used the five skill types as our KCs, and tracked students' learning progress of these skills by Bayesian Knowledge Tracing (BKT) [60]. The BKT parameters were set as $p(L_0) = 0.4$, $p(T) = 0.05$, $p(S) = p(G) = 0.299$ [3], and the mastery threshold is 0.9.

First, we looked at how well students mastered each of the five skills in the game. Comparing the students' final mastery probabilities in each skill and our mastery threshold,

we observed that: there were 4 students who did not master any skill, 20 students who mastered one skill, 33 students who mastered two skills, 42 students who mastered three skills, 34 students who mastered four skills, and 26 students who mastered all five skills. Next, we counted how many opportunities each student who mastered a skill took to reach mastery in that skill. An opportunity is defined as one complete decimal exercise; each mini-game round consists of one opportunity, except for those in **Sequence**, which contain three opportunities (i.e., students have to fill in three decimal sequences per round). The distributions of opportunity count until mastery are plotted in Figure 2, which shows that **Number Line** and **Sorting** took the longest to master, at around 5 opportunities on average. For **Number Line**, one student even needed 26 opportunities to reach mastery.

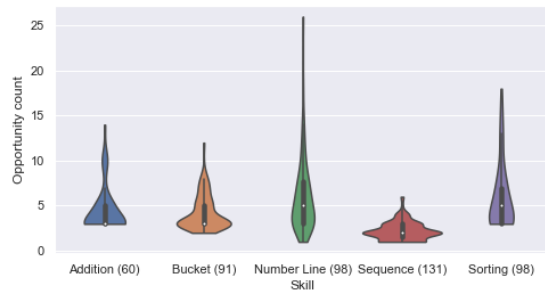


Figure 2: Opportunity counts until mastery for each skill. The number next to each skill indicates the count of students who mastered that skill and were included in the violin plot.

Next, we examined how well students regulated their learning, i.e., after mastering a skill, did they tend to continue practicing the same skill, or switch to a different skill? For each student, following [8], once they mastered a skill ($\geq 90\%$ mastery probability), we considered their subsequent opportunities as over-practice. Then, for each student who mastered a particular skill, we computed the ratio between their over-practice count and total opportunity count in that skill. Plotting these ratios for all the mastered students in each skill (Figure 3), we observed that between 20-80% of a student’s practice opportunities in a skill could be considered over-practice, i.e., they took place after the student had mastered the skill.

4.2 Investigating factors related to posttest and delayed posttest performance

Having examined students’ in-game learning, we then looked at how it related to test performance after the game. In order to predict posttest and delayed posttest scores, we collected features that reflected students’ in-game learning and also included demographic measures that account for individual student differences. In total, we considered 19 features: pretest score, decimal efficacy, gender, computer efficacy, identification agency, intrinsic agency, external agency, perseverance, utility, math interest, expectancy, final in-game mastery probabilities of the five skills (**Addition**, **Bucket**, **Sequence**, **Number Line**, **Sorting**), total opportunity count, over-practice opportunity count and total incorrect answer counts. To identify the most important features, we (1) per-

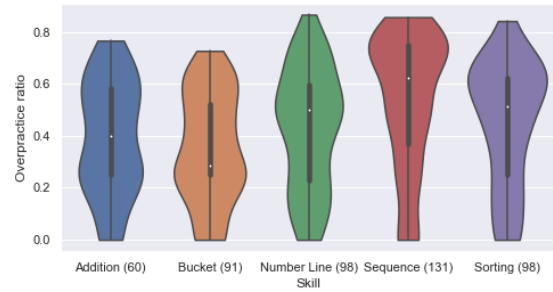


Figure 3: Over-practice ratio in each skill. The number next to each skill indicates the count of students who mastered that skill and were included in the violin plot.

formed feature selection with linear regression, and (2) ran another linear regression model with the selected features on the full dataset to inspect the coefficient and significance of each feature. In step (1), we use the `mlxtend` library [45] to run a forward feature selection procedure that returns the feature subset with the best cross-validated performance, measured in terms of mean squared error (MSE).

In predicting posttest scores, our feature selection identified three features: **Bucket** mastery, **Sorting** mastery and pretest score. A linear regression model with these three features, when trained and evaluated on the entire dataset, had an MSE of 26.167 and an adjusted R^2 of .735. Based on the regression table, the coefficient and significance of each feature was as follow: pretest score with $\beta = 0.734$, $p < .001$, **Bucket** mastery with $\beta = 6.833$, $p < .001$, **Sorting** mastery with $\beta = 5.100$, $p = .001$. In other words, pretest scores, **Bucket** mastery and **Sorting** mastery each had a positive and significant association with posttest scores.

The delayed posttest model incorporated two additional features – **Number Line** mastery and gender – and yielded an MSE of 24.218, as well as an adjusted R^2 of .747. Based on the regression table, the coefficient and significance of each feature was as follows: pretest score with $\beta = 0.730$, $p < .001$, **Bucket** mastery with $\beta = 4.276$, $p = .018$, **Sorting** mastery with $\beta = 4.270$, $p = .003$, **Number Line** with $\beta = 3.099$, $p = .029$, and gender with $\beta = 1.426$, $p = .074$. In other words, the three skill mastery values – **Bucket**, **Sorting**, **Number Line** – as well as pretest score each had a positive and significant association with delayed posttest score, while gender (male = 0, female = 1) had a positive and marginally significant association.

4.3 Investigating factors related to enjoyment

For each enjoyment dimension measured in post-intervention surveys (achievement emotion, game engagement, affective engagement - see Table 1), we computed the per-student average Likert scores to the statements in that dimension. Then, we performed the same feature selection procedure as in 4.2 and reported our results in Table 2.

We observed that the adjusted R^2 values of the game engagement and affective engagement models were much lower

Table 2: Results of feature selection for predicting game enjoyment. The Overall performance row indicates the selected model’s scores when trained and evaluated on the entire dataset.

	Achievement Emotion	Game Engagement	Affective Engagement
Selected features	computer efficacy, identification agency, intrinsic agency, math interest, pretest score, total opportunity count	math interest, computer efficacy, gender	decimal efficacy, gender, intrinsic agency, Sorting mastery, Bucket mastery, total incorrect attempt count, identification agency
Overall	MSE = 0.520	MSE = 0.602	MSE = 0.660
performance	Adjusted R^2 = 0.386	Adjusted R^2 = 0.225	Adjusted R^2 = 0.218

than those of the test score models. Even when trained and evaluated on the entire dataset, Linear Regression could only explain about 20% of the variance in game engagement and affective engagement. On the other hand, the achievement emotion model did have reasonable performance (adjusted R^2 = .386), so we focused on analyzing the features in this model. The linear regression table showed the coefficient and significance of each feature as follows: computer efficacy with β = 0.047, p = .063, identification agency with β = 0.099, p = .024, intrinsic agency with β = 0.116, p = .002, math interest with β = 0.114, p = .001, pretest score with β = -0.017, p = .011, opportunity count with β = 0.009, p = .033. In other words, computer efficacy had a positive and marginally significant association, while pretest score had a negative and significant association; the remaining features (identification agency, intrinsic agency, math interest and opportunity count) each had a positive and significant association.

5. DISCUSSION

5.1 Investigating in-game learning

Based on the opportunity count until mastery in each skill (Figure 2), we identified **Sorting** and **Number Line** as the most difficult skills in the game. Our prior learning curve analysis [40] on a different *Decimal Point* study reported a consistent finding – that the learning curves of these two skills were mostly flat and reflected small learning rates. Based on previous research in decimal learning, a plausible explanation is that there are several misconceptions which can lead to students making a mistake in **Sorting** or **Number Line** problems, including (1) treating decimals as whole numbers, (2) treating decimals as fractions, and (3) ignoring the zero in the tenths place [46]. Furthermore, even when students recognize their misconception, they may shift to a different misconception instead of arriving at the correct understanding [56]. This phenomenon likely also occurred in *Decimal Point*, as the game provides corrective feedback (whether an answer is right or wrong) but does not emphasize the underlying reasoning; consequently, as an example, a student realizing it is wrong to assume longer decimals are larger may end up concluding that shorter decimals must be larger, thereby adopting a new misconception. This highlights the need for more refined tracing of the student’s dynamic learning states in a digital learning environment. While the standard KC modeling technique can track when students make an intended mistake (e.g., longer decimals are larger), it does not investigate their specific input to see whether a new misconception (e.g., shorter decimals are larger) has emerged. To address this issue, future itera-

tions of the game should provide more instructional support that can react to various misconceptions from students, for example via explanatory feedback [19] or predefined error messages for different types of error [36].

Once students have mastered a skill, however, our analysis showed that over-practice was very common, i.e., students kept playing more mini-games in the mastered skill. At the same time, there were only 26 out of 159 students who mastered all five skills, suggesting that the majority of students still had room for improvement in the unmastered skills but chose not to practice them. One possible reason is that the game environment did not explicitly indicate when the student has reached mastery or force them to switch to practicing a different skill. Consequently, young students, who were likely to be weak at self-regulated learning [37,53], simply played the mini-games that they thought were engaging, which in this case involved the skills they had already mastered. A prior study by [29] similarly found that, in a game about locating fractions on number line, students were more engaged when the game was easier, contradicting game design theories that optimal engagement would occur at moderate difficulty level.

5.2 Investigating factors related to posttest and delayed posttest performance

We saw that our linear regression models were able to predict posttest and delayed posttest performance well, capturing about 75% of the variance in test scores with only 3-5 features. The three features present in both models are pretest score, **Sorting** mastery and **Bucket** mastery. The inclusion of pretest score is not surprising, as it is consistent with the standard practice of controlling for prior knowledge when analyzing posttest score [58]. On the other hand, both **Sorting** mastery and **Bucket** mastery suggest that the ability to compare decimal numbers plays a large role in test performance. This is likely due to the game and test materials focusing on the four most common decimal misconceptions (Megz, Segz, Pegz, Negz), three of which are related to decimal comparison [25]. Based on the distribution of practice opportunities until mastery, however, students took much more attempts to master **Sorting** problems than **Bucket** problems, which may explain why they did not achieve high scores on the posttest and delayed posttest, averaging at only around 30 out of 52 points [22]. Therefore, improving students’ performance on **Sorting** problems, potentially by incorporating hints and error messages as we previously discussed, is crucial in future studies of the game.

At the same time, we saw that **Number Line** mastery had a significant positive association with delayed posttest score, but was not selected in the posttest model. An interpretation of this result is that **Number Line** tasks, which we identified as among the most difficult in the game, could be at a *desirable difficulty* level, which can promote deeper and longer-lasting learning than the more straightforward tasks [61]. For instance, a prior study on comparing erroneous examples and problem-solving decimal tasks found that erroneous examples, which are more aligned with the desirable difficulty, led to significantly higher delayed posttest scores but similar posttest scores [34]. In our case, we also saw that **Number Line** is an important feature for predicting delayed posttest but not for predicting posttest performance.

Similar to **Number Line** mastery, gender (male = 0, female = 1) was not a feature in the posttest model, but had a positive association with delayed posttest scores. In other words, with other factors being equal, females could achieve higher delayed posttest scores than males. While this association is only marginally significant ($p = .074$), similar findings about females' tendency to outperform males in retention and delayed posttest have been reported in previous mathematics intervention studies [1, 20]. Using the same dataset as in this work, [22] also found that females demonstrated significantly higher pre-post and pre-delayed learning gains than males, with a larger effect size in pre-delayed learning gains. Therefore, an important next step is to conduct future studies of *Decimal Point* on a larger sample size to draw more conclusive findings about whether the game promotes more retention in females and what could lead to this effect.

5.3 Investigating factors related to enjoyment

Our enjoyment prediction models did not perform as well as the learning models and could explain only about 20% of the variance in game engagement and affective engagement. These poor model fits likely result from the lack of appropriate features in our data. To track student engagement, previous work has emphasized the use of fine-grained measures such as time spent on decision making [47], social engagement profile [49] and interaction traces [6]; in contrast, our feature set consists mainly of quantitative scores (e.g., Likert responses) and aggregate data (e.g., error count). Related to this direction, a previous study of *Decimal Point* by [57] has clustered students based on their mini-game selection orders and found that the cluster which demonstrated more agency reported higher enjoyment. Adopting their method of encoding students' mini-game sequences is a good first step in building more fine-grained features for our prediction tasks. On the other hand, the lack of association between our in-game learning measures (e.g., skill mastery, over-practice opportunity count, error count) and game engagement or affective engagement implies that students' game performance, whether good or bad, were unlikely to yield any negative emotion such as confusion or frustration. This is a positive outcome, indicating that our game environment does not impose any performance pressure on students – one of the primary principles of learning games [15].

At the same time, we did find that a linear regression model was able to predict achievement emotion reasonably well from student's identification agency, intrinsic agency, math interest, computer efficacy, pretest score and opportunity

count. Identification and intrinsic agency indicate that, with all other factors being equal, the more students identified their learning as coming from intrinsic motivation (rather than external pressures), the more achievement they felt after learning. Math interest and computer efficacy suggest that students' acquaintance with the learning domain or medium could also be positively associated with achievement emotion [26]. On the other hand, pretest score had a negative association, likely because students with lower prior knowledge were able to learn more from the game and therefore felt more achievement than those with high prior knowledge. Similarly, for opportunity count, a plausible reason for students choosing to play more mini-game rounds is that they felt the mini-games were helpful, which contributed to their achievement emotion after game play. Overall, the features we identified could serve as a guideline for promoting achievement emotion in learning games and in more general instructional contexts.

6. CONCLUSIONS

From our analyses, we gained several insights into students' learning outcomes and enjoyment in *Decimal Point*. First, we found that **Sorting** and **Number Line** are important skills for posttest and delayed posttest performance, but students required more instructional support to effectively master them. Second, very few students mastered all five decimal skills from the game, while the majority engaged in over-practice, likely due to their preference for playing easy mini-games, i.e., those they had already mastered. Third, expanding on prior findings about gender effect in *Decimal Point* [22, 33], we identified a trend of females outperforming males in the delayed posttest, which should be investigated on a larger sample size. Fourth, we learned that students' achievement emotion can be reasonably captured by their level of computer efficacy, learning motivation, prior knowledge and number of mini-game rounds. All of these insights can be derived from log data alone and would serve as useful metrics to assist digital learning game researchers in evaluating and improving their own games. For *Decimal Point*, in particular, an important next step is to perform similar analyses in other studies of the game to see which of our findings can be replicated. Identifying consistent trends in student data could allow us to construct a more generalized model of students' game play that combines existing theories with novel exploratory analyses [38].

In a broader context, we have seen the rapid growth of digital learning games in recent years, from being conceived as a novel learning platform [15, 21] to having their effectiveness validated by rigorous studies [10]. The game *Decimal Point*, in particular, has been shown to significantly improve students' learning through several research works [18, 22, 35, 39]. When viewing from a learning analytics perspective, however, one could identify room for improvement that would otherwise not be reflected in pretest and posttest scores alone. For instance, a game may not adequately support all of its learning objectives, or students may engage in non-optimal learning behavior due to a lack of self-regulation. At the heart of these issues is the question of how digital learning games can optimize student learning while retaining its core value as a playful environment, where players are free to exercise their agency. Addressing this question is an important step for future works in the field.

7. REFERENCES

- [1] J. Ajai and B. Imoko. Gender differences in mathematics achievement and retention scores: A case of problem-based learning method. *International Journal of research in Education and Science*, 1(1):45–50, 2015.
- [2] E. Andersen, E. O’Rourke, Y.-E. Liu, R. Snider, J. Lowdermilk, D. Truong, S. Cooper, and Z. Popovic. The impact of tutorials on games of varying complexity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 59–68, 2012.
- [3] R. S. Baker. Personal correspondence, 2019.
- [4] R. S. Baker, M. J. Habgood, S. E. Ainsworth, and A. T. Corbett. Modeling the acquisition of fluent skill in educational action games. In *International Conference on User Modeling*, pages 17–26. Springer, 2007.
- [5] A. Ben-Eliyahu, D. Moore, R. Dorph, and C. D. Schunn. Investigating the multidimensionality of engagement: Affective, behavioral, and cognitive engagement across science activities and contexts. *Contemporary Educational Psychology*, 53:87–105, 2018.
- [6] P. Bouvier, K. Sehaba, and É. Lavoué. A trace-based approach to identifying users’ engagement and qualifying their engaged-behaviours in interactive systems: application to a social game. *User Modeling and User-Adapted Interaction*, 24(5):413–451, 2014.
- [7] J. H. Brockmyer, C. M. Fox, K. A. Curtiss, E. McBroom, K. M. Burkhart, and J. N. Pidruzny. The development of the game engagement questionnaire: A measure of engagement in video game-playing. *Journal of Experimental Social Psychology*, 45(4):624–634, 2009.
- [8] H. Cen, K. R. Koedinger, and B. Junker. Is over practice necessary?-improving learning efficiency with the cognitive tutor through educational data mining. *Frontiers in artificial intelligence and applications*, 158:511, 2007.
- [9] C.-H. Chen, K.-C. Wang, and Y.-H. Lin. The comparison of solitary and collaborative modes of game-based learning on students’ science learning and motivation. *Journal of Educational Technology & Society*, 18(2):237–248, 2015.
- [10] D. B. Clark, E. E. Tanner-Smith, and S. S. Killingsworth. Digital games, design, and learning: A systematic review and meta-analysis. *Review of educational research*, 86(1):79–122, 2016.
- [11] G. C. Delacruz, G. K. Chung, and E. L. Baker. Validity evidence for games as assessment environments. cress report 773. *National Center for Research on Evaluation, Standards, and Student Testing (CRESST)*, 2010.
- [12] A. L. Duckworth, C. Peterson, M. D. Matthews, and D. R. Kelly. Grit: perseverance and passion for long-term goals. *Journal of personality and social psychology*, 92(6):1087, 2007.
- [13] A. M. Durik, M. Vida, and J. S. Eccles. Task values and ability beliefs as predictors of high school literacy choices: A developmental analysis. *Journal of Educational Psychology*, 98(2):382, 2006.
- [14] W. Fan and C. A. Wolters. School motivation and high school dropout: The mediating role of educational expectation. *British Journal of Educational Psychology*, 84(1):22–39, 2014.
- [15] J. P. Gee. What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, 1(1):20–20, 2003.
- [16] J. F. Hair, W. C. Black, B. J. Babin, R. E. Anderson, R. L. Tatham, et al. Multivariate data analysis (vol. 6), 2006.
- [17] E. Harpstead and V. Aleven. Using empirical learning curve analysis to inform design in an educational game. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, pages 197–207, 2015.
- [18] E. Harpstead, J. E. Richey, H. Nguyen, and B. M. McLaren. Exploring the subtleties of agency and indirect control in digital learning games. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 121–129, 2019.
- [19] J. Hattie and H. Timperley. The power of feedback. *Review of educational research*, 77(1):81–112, 2007.
- [20] L. L. Haynes and J. V. Dempsey. How and why students play computer-based mathematics games: A consideration of gender differences. *2001 Annual Proceedings-Atlanta: Volume*, page 178.
- [21] M. A. Honey and M. L. Hilton. Learning science through computer games. *National Academies Press, Washington, DC*, 2011.
- [22] X. Hou, H. Nguyen, J. E. Richey, and B. M. McLaren. Exploring how gender and enjoyment impact learning in a digital learning game. In *International Conference on Artificial Intelligence in Education*. Springer, 2020.
- [23] C. S. Hulleman, O. Godes, B. L. Hendricks, and J. M. Harackiewicz. Enhancing interest and performance with a utility value intervention. *Journal of educational psychology*, 102(4):880, 2010.
- [24] A. Illanas Vila, J. R. Calvo-Ferrer, F. J. Gallego-Durán, F. Llorens Largo, et al. Predicting student performance in foreign languages with a serious game. 2013.
- [25] S. Isotani, D. Adams, R. E. Mayer, K. Durkin, B. Rittle-Johnson, and B. M. McLaren. Can erroneous examples help middle-school students learn decimals? In *European Conference on Technology Enhanced Learning*, pages 181–195. Springer, 2011.
- [26] M. Jansen, O. Lüdtke, and U. Schroeders. Evidence for a positive relation between interest and achievement: Examining between-person and within-person variation in five domains. *Contemporary Educational Psychology*, 46:116–127, 2016.
- [27] S. Karumbaiah, R. S. Baker, and V. Shute. Predicting quitting in students playing a learning game. *International Educational Data Mining Society*, 2018.
- [28] K. R. Koedinger, E. Brunskill, R. S. Baker, E. A. McLaughlin, and J. Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 34(3):27–41, 2013.
- [29] D. Lomas, K. Patel, J. L. Forlizzi, and K. R. Koedinger. Optimizing challenge in an educational game using large-scale design experiments. In *Proceedings of the SIGCHI Conference on Human*

- Factors in Computing Systems*, pages 89–98, 2013.
- [30] M. Manske and C. Conati. Modelling learning in an educational game. In *AIED*, pages 411–418, 2005.
 - [31] G. Marakas, R. Johnson, and P. F. Clay. The evolving nature of the computer self-efficacy construct: An empirical investigation of measurement construction, validity, reliability and stability over time. *Journal of the Association for Information Systems*, 8(1):2, 2007.
 - [32] R. E. Mayer. *Computer games for learning: An evidence-based approach*. MIT Press, 2014.
 - [33] B. McLaren, R. Farzan, D. Adams, R. Mayer, and J. Forlizzi. Uncovering gender and problem difficulty effects in learning with an educational game. In *International Conference on Artificial Intelligence in Education*, pages 540–543. Springer, 2017.
 - [34] B. M. McLaren, D. M. Adams, and R. E. Mayer. Delayed learning effects with erroneous examples: a study of learning decimals with a web-based tutor. *International Journal of Artificial Intelligence in Education*, 25(4):520–542, 2015.
 - [35] B. M. McLaren, D. M. Adams, R. E. Mayer, and J. Forlizzi. A computer-based game that promotes mathematics learning more than a conventional approach. *International Journal of Game-Based Learning (IJGBL)*, 7(1):36–56, 2017.
 - [36] B. M. McLaren, S.-J. Lim, D. Yaron, and K. R. Koedinger. Can a polite intelligent tutoring system lead to improved learning outside of the lab? *Frontiers in Artificial Intelligence and Applications*, 158:433, 2007.
 - [37] J. Metcalfe and N. Kornell. The dynamics of learning and allocation of study time to a region of proximal learning. *Journal of Experimental Psychology: General*, 132(4):530, 2003.
 - [38] R. J. Mislevy, J. T. Behrens, K. E. Dicerbo, and R. Levy. Design and discovery in educational assessment: Evidence-centered design, psychometrics, and educational data mining. *JEDM| Journal of Educational Data Mining*, 4(1):11–48, 2012.
 - [39] H. Nguyen, E. Harpstead, Y. Wang, and B. M. McLaren. Student agency and game-based learning: A study comparing low and high agency. In *International Conference on Artificial Intelligence in Education*, pages 338–351. Springer, 2018.
 - [40] H. Nguyen, Y. Wang, J. Stamper, and B. M. McLaren. Using knowledge component modeling to increase domain understanding in a digital learning game. In *International Conference on Educational Data Mining*, pages 139–148, 2019.
 - [41] M. Ninaus, K. Moeller, J. McMullen, and K. Kiili. Acceptance of game-based learning and intrinsic motivation as predictors for learning success and flow experience. 2017.
 - [42] Z. Peddycord-Liu, R. Harred, S. Karamarkovich, T. Barnes, C. Lynch, and T. Rutherford. Learning curve analysis in a large-scale, drill-and-practice serious math game: Where is learning support needed? In *International Conference on Artificial Intelligence in Education*, pages 436–449. Springer, 2018.
 - [43] R. Pekrun. Progress and open problems in educational emotion research. *Learning and Instruction*, 15(5):497–506, 2005.
 - [44] P. Pintrich, D. Smith, T. Garcia, and W. McKeachie. A manual for the use of the motivated strategies for learning questionnaire (mslq) ann arbor. *MI: National Center for Research to Improve Postsecondary Teaching and Learning*, pages 1–76, 1991.
 - [45] S. Raschka. Mlxtend: Providing machine learning and data science utilities and extensions to python’s scientific computing stack. *The Journal of Open Source Software*, 3(24), Apr. 2018.
 - [46] L. B. Resnick, P. Nesher, F. Leonard, M. Magone, S. Omanson, and I. Peled. Conceptual bases of arithmetic errors: The case of decimal fractions. *Journal for research in mathematics education*, pages 8–27, 1989.
 - [47] V. Riemer and C. Schrader. Impacts of behavioral engagement and self-monitoring on the development of mental models through serious games: Inferences from in-game measures. *Computers in Human Behavior*, 64:264–273, 2016.
 - [48] J. P. Rowe and J. C. Lester. Modeling user knowledge with dynamic bayesian networks in interactive narrative environments. In *Sixth AI and Interactive Digital Entertainment Conference*, 2010.
 - [49] J. A. Ruiperez-Valiente, M. Gaydos, L. Rosenheck, Y. J. Kim, and E. Klopfer. Patterns of engagement in an educational massive multiplayer online game: A multidimensional view. *IEEE Transactions on Learning Technologies*, 2020.
 - [50] R. M. Ryan and J. P. Connell. Perceived locus of causality and internalization: Examining reasons for acting in two domains. *Journal of personality and social psychology*, 57(5):749, 1989.
 - [51] J. L. Sabourin, L. R. Shores, B. W. Mott, and J. C. Lester. Understanding and predicting student self-regulated learning strategies in game-based learning environments. *International Journal of Artificial Intelligence in Education*, 23(1-4):94–114, 2013.
 - [52] R. Sawyer, A. Smith, J. Rowe, R. Azevedo, and J. Lester. Is more agency better? the impact of student agency on game-based learning. In *International Conference on Artificial Intelligence in Education*, pages 335–346. Springer, 2017.
 - [53] W. Schneider. The development of metacognitive knowledge in children and adolescents: Major trends and implications for education. *Mind, Brain, and Education*, 2(3):114–121, 2008.
 - [54] V. J. Shute, L. Wang, S. Greiff, W. Zhao, and G. Moore. Measuring problem solving skills via stealth assessment in an engaging video game. *Computers in Human Behavior*, 63:106–117, 2016.
 - [55] J. Stamper, K. Koedinger, R. S. d Baker, A. Skogsholm, B. Leber, J. Rankin, and S. Demi. Pslc datashop: A data analysis service for the learning science community. In *International Conference on Intelligent Tutoring Systems*, pages 455–455. Springer, 2010.
 - [56] W. Van Dooren, D. De Bock, A. Hessels, D. Janssens, and L. Verschaffel. Remedying secondary school students’ illusion of linearity: A teaching experiment aiming at conceptual change. *Learning and Instruction*, 14(5):485–501, 2004.

- [57] Y. Wang, H. Nguyen, E. Harpstead, J. Stamper, and B. M. McLaren. How does order of gameplay impact learning and enjoyment in a digital learning game? In *International Conference on Artificial Intelligence in Education*, pages 518–531. Springer, 2019.
- [58] B. E. Whitley and M. E. Kite. *Principles of research in behavioral science*. Routledge, 2013.
- [59] J. Wiemeyer, M. Kickmeier-Rust, and C. M. Steiner. Performance assessment in serious games. In *Serious Games*, pages 273–302. Springer, 2016.
- [60] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *International conference on artificial intelligence in education*, pages 171–180. Springer, 2013.
- [61] C. L. Yue, E. L. Bjork, and R. A. Bjork. Reducing verbal redundancy in multimedia learning: An undesired desirable difficulty? *Journal of Educational Psychology*, 105(2):266, 2013.

Measuring Ability-to-Learn Using Parametric Learning-Gain Functions

Chris Piech, Engin Bumbacher, Richard Davis
Stanford University
piech@cs.stanford.edu, rldavis@stanford.edu

ABSTRACT

One crucial function of a classroom, and a school more generally, is to prepare students for future learning. Students should have the capacity to learn new information and to acquire new skills. This ability to “learn” is a core competency in our rapidly changing world. But how do we measure ability to learn? And how can we measure how well a school has prepared their students to learn? In this paper we formally pose the problem, and introduce a grounded theory of how to measure ability to learn. Using simulations of students learning we provide initial evidence that this theory provides an elegant solution to this problem. We further validate our ideas using real world data from 70k middle-school students and show that our theory is more accurate and interpretable than current state-of-the-art models of learning gains. We consider our results a modest yet interesting first step for a novel type of test.

1. INTRODUCTION

Large-scale, standardized tests typically measure knowledge and skills that students already possess, such as reading comprehension and mathematical competency. However, these tests overlook students’ abilities to acquire new knowledge and skills. Could we instead measure how well a student is able to *learn*? Measuring how well a school system has prepared a student for learning is a particularly hard challenge and as such it remains elusive. PISA (Programme for International Student Assessment – an international test run every three years to evaluate educational systems), has made it a goal of their 2024 innovative assessment to measure ability to learn. How could such a test be scored?

Early research has shown that measuring ability to learn is both important and difficult. Work by Schwartz et al. [18, 17, 19] has shown that assessments of students’ ability to learn capture important information that assessments which simply measure what a student knows fail to capture. In these studies, students participated in two different educational interventions, one designed to teach students factual content

in a manner that also prepared them for future learning, and one designed to teach students factual content using more traditional approaches. Standard measures of knowledge found that regardless of the intervention, students in both groups learned the same factual content. However, a second type of assessment designed to measure students’ ability to learn uncovered significant predictive differences.

Despite the potential, to the best of our knowledge, there are no large-scale assessment that have attempted to measure students’ ability to learn. In traditional tests, students get questions correct or incorrect — a single random variable that is traditionally modeled using Item-Response Theory (IRT). In a learning test, on the other hand, students work through learning experiences which produce two measurable values: a **prior (pre)** and **posterior (post)** ability. All learning experiences, especially relevant authentic ones, are impacted by what a student knows when they start. A useful model would enable measurement of student learning across countries, schools-districts, and millions of students as they engage in a necessarily wide variety of learning experiences. Without a useful model it is hard (if not impossible) to produce desired and important analyses such as: (a) inferring ability to learn from multiple learning experiences (b) discovering issues of fairness in learning experiences (c) reasoning about mixture effects within populations.

The prior-knowledge confound: Measuring learning-ability is particularly difficult because it requires us to reason about the impact of prior knowledge. For example, consider two populations where students have the exact same ability to learn but different levels of prior knowledge. Now imagine the two populations are given the same learning experience. Both populations will learn (recall they have the same learning ability) but will have different outcomes on the same exam. In practice most people model this relationship using a “linear” model [22]. However, research has shown that the impact of prior ability has important non linear properties [21, 15]. This is an instance of Simpson’s Paradox.

A core insight of this paper is to think of the difference between prior and posterior ability as being governed by population specific parametric functions which we call **Learning-Gain Functions**. These learning gain functions are naturally incorporated in a fully Bayesian model of student responses on learning ability tests. The main contributions of this paper are:

1. We formalize, parametric Learning-Gain functions as a

Chris Piech, Engin Bumbacher and Richard Davis "Measuring Ability-to-Learn Using Parametric Learning Gain Functions" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 496 - 502

way to model ability-to-learn tests.

2. We introduce an interpretable single-parameter Bayesian family of Learning-Gain functions.
3. We show that this model is able to near-perfectly recover learning ability in a complex, simulated dataset.
4. We demonstrate that this model outperforms other single- and multi-parameter models on two real-world datasets.
5. We show the practical value of this model by comparing real-world schools on their “ability to learn” as estimated by our model

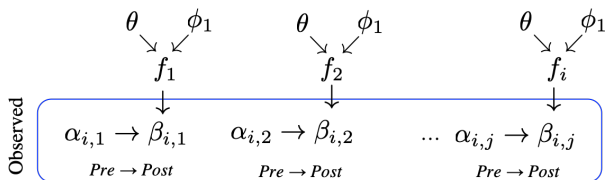
This work is a first attempt at addressing the need for models of student “ability to learn” that can be employed in large-scale assessments such as PISA 2024. The initial results are promising, and we hypothesize that the model will generalize broadly to different learning tests.

1.1 Population Learning-Ability Tests

Learning-ability-tests are built to directly measure the “ability to learn” of a population. The most straightforward format for such an exam has learners complete a set of learning tasks and for each task j , the learner i is given a pre and post test – these fence-post the learning gains. We define alpha ($\alpha_{i,j}$) to be a student’s prior ability on the task and beta ($\beta_{i,j}$) to be the student’s posterior ability.

We seek to measure ability-to-learn for a population (or an individual as a singleton population) as a number, which we call θ . This measure should generalize and explain ability-to-learn of the population on a different learning-task. In order to learn a generalizable θ we must learn to separate ability-to-learn from task specific effects (such as if the task is easier for beginners to learn than for advanced students etc). We use the notation phi (ϕ_j) to represent task specific parameters for task j .

We propose that when a student engages with a learning task, the learning-ability of the student (θ) interacts with task-specific-parameters (ϕ_j) to produce a **learning-gain-function** (f_j) which determines how prior-abilities will map to post-abilities. As such a function oriented probabilistic model of a single student, from a population with learning-ability θ , working on a series of learning tasks would look like the following:



Learning-ability tests stand in contrast to Intelligence Quotient (IQ) exams as measurement takes place on either end of a learning experience. IQ tests on the other hand measure aptitude, and while this often requires learners to engage in complex tasks the goal is to measure ability on the task.

1.2 Prior work

This work builds on a rich and broad literature of work on measuring ability-to-learn which extends for decades [5, 13, 9]. Evaluation of students’ ability to learn is often treated as equivalent with change in knowledge over time, typically with a pretest and posttest. Common approaches include comparison of raw gain scores (posttest minus pretest), analysis of posttest scores with pretest scores as a covariate, and analysis of gains scores with pretest scores as a covariate. Each of these methods has strengths and weaknesses, although there is evidence that analysis of gain scores with pretest scores as a covariate is the best of these methods when certain assumptions are met [6]. As such, we included this model (Linear Multi-Theta) in our model comparison on real-world data and find that it doesn’t fit as well. Additionally, while the intercept and slope parameters in the Linear Multi-Theta model can be interpreted as describing a population’s ability to learn, it is not immediately clear how they might be used to compare different populations. Both the Learning-Gain-Decay and Learning-Gain-Bump models estimate ability to learn with a single parameter, avoiding this problem. Taken together, these factors suggest that it would be prudent to move away from the Linear Multi-Theta model if our goal is to estimate ability to learn.

Another approach to estimating student ability to learn is to characterize “learning curves” [7]. This requires repeated sampling over time so the learning rate can be determined from the shape of the curve, where students with higher ability to learn are characterized by steeper learning curves, and students with lower ability to learn are characterized by shallower curves. However, the shape of a learning curve does not reveal the full interaction between prior and posterior knowledge. We would expect two students with the same ability to learn but different levels of prior knowledge to progress at significantly different rates. Additionally, collecting enough data to plot a learning curve requires repeated measurement that is infeasible in most educational settings.

NWEA has looked into how to quantify learning gains [12, 11] and most recently [21]. Their contemporary models project student abilities into norm grade levels. [16, 8]. Anderman et al make initial steps into translating learning-gain research into a bayesian model [1]

Significant research has focused on the promise and perils of using student gain data as an outcome—as a good indicator of teacher effectiveness. There is a book on the subject of evaluating teachers by measuring their value added: Evaluating Value-Added Models for Teacher Accountability [10]. We remind the reader that it is necessary to be careful and accurate in measuring student learning.

There is a rich mathematical history of reasoning about functional mappings. This field of mathematics draws from domains as diverse as 3D geometry [14, 3] to neocortical circuitry [20]. This is, to the best of our knowledge, the first use of functional maps in measuring learning.

1.3 Learning Gain Functions

In traditional IRT, each interaction between a student and a question (aka item) produces a single number. In a learning test, each learning-experience produces two numbers ($\alpha_{i,j}$

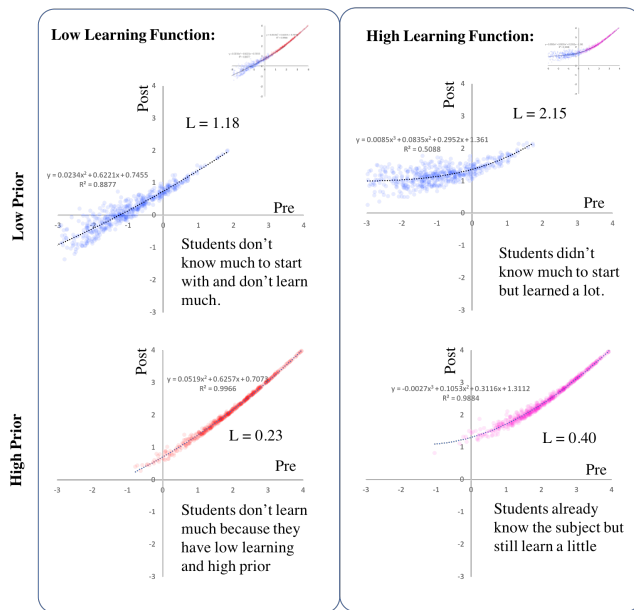


Figure 1: Simulations of four populations on the same task. Each graph represents pre/post abilities for one population. Each point represents one student. Countries in the columns have the same learning function. $L = \mu$ post minus pre.

and $\beta_{i,j}$). This poses a modelling challenge. How do we model learning, in a way that elegantly considers the effect of prior knowledge ($\alpha_{i,j}$)?

We found it natural to resolve this problem by thinking of the learning experience as being a reflection of an underlying *function* which we call learning-gain-function. A learning-gain-function is a population wide mapping of student pre-conditions to post-conditions. In a learning-ability-test, we would like to compare populations on their ability to learn, and as such it seems like it would be best to compare the countries by their functional mapping. Thinking of the mapping of prior-ability to post-ability is incredibly useful if we want to build a Bayesian model of learning-ability.

To articulate this point, consider the four different populations learning on the exact same learning task (Figure 1). For all four populations we plot prior abilities and posterior abilities. The two populations on the left both have the same learning “function” on this task. If you had two students with the same prior ability, after the learning-task they would have (within noise) the same post ability. As a confound, they have different prior ability distributions. Typical measures of learning gains would compare these two populations based on the average difference in post ability versus pre ability (L shown on the figure). *As such they would look very different even though the two populations have the same learning-gain-function.* The same is true for the two populations on the right. They also have the exact same learning-gain-function, but as a result of different prior knowledge distributions, typical metrics make them seem quite different. By modelling a learning-gain-function we neither benefit, nor penalize populations for having different prior distributions. Instead we compare learning in a way that is agnostic to previous knowledge.

The learning function f is “parameterized” by the ability-to-learn parameter θ and task specific parameters, ϕ :

$$f_{\theta_i, \phi_j}(\alpha_{i,j}) \rightarrow \beta_{i,j}$$

In the case of PISA, this theta should represent “ability to learn” for a specific population. The function, importantly, does not have to be linear – and in fact ample evidence shows that it should not be. Note that, $\alpha_{i,j}$ and $\beta_{i,j}$ can be estimated using standard item-response theory.

This formalization lends some insight into how we can deal with the different levels of prior knowledge between populations. At this point we haven’t made any claims about what the function looks like. What is an appropriate parametric form of a learning-gain function?

2. SIMULATING LEARNING

To begin the process of understanding the family of functions for how much students learn during a task, we built a series of simulators in python that attempt to match as realistically as possible the process of learning during a task. The simulator has fake students learn through the process of working on fake items, where the learning and progress at each minute is governed by the interaction between a student’s prior knowledge and the difficulty of the items (an assumption loosely based on the zone of proximal development). This simulation is not perfect, but it provides us with a starting point for building a theory of ability to learn. It is simple, and makes it possible to observe all the factors that impact changes in knowledge, including variables which are often unobservable like learning ability.

These simulations have the added benefit of building a falsifiable condition for any model which tries to estimate ability to learn. I.e., any good model should be able to describe this synthetic data. While ability to describe synthetic data is evidence in support of a theory, it is a necessary but not sufficient condition. The final test would be to show that it also works with real world data.

Figure 1 shows a simulation of 2,000 students learning in four countries via a single task which is heavily biased towards “beginners learn more”. The countries in the right column both have the exact same learning function, but because their students have different priors, they are very hard to distinguish that they have the same learning ability.

The main take away at this point is to confirm what we believed from prior work: average “ability” gain is not a very useful metric. Even for countries with the exact same learning rate we observe very different average gains (L) when priors are different.

3. THEORY OF LEARNING FUNCTIONS

If we could come up with an equation for that function (aka the form) we could formalize our measurement of ability-to-learn. In the example from Figure 1 it feels like a “polynomial” fits f well – but that turns out to be a bit misleading. The learning-experience in that figure represents one where “beginners learn more.” If we change the learning task to be one where “medium level students learn more” the function is not well fit by a polynomial.

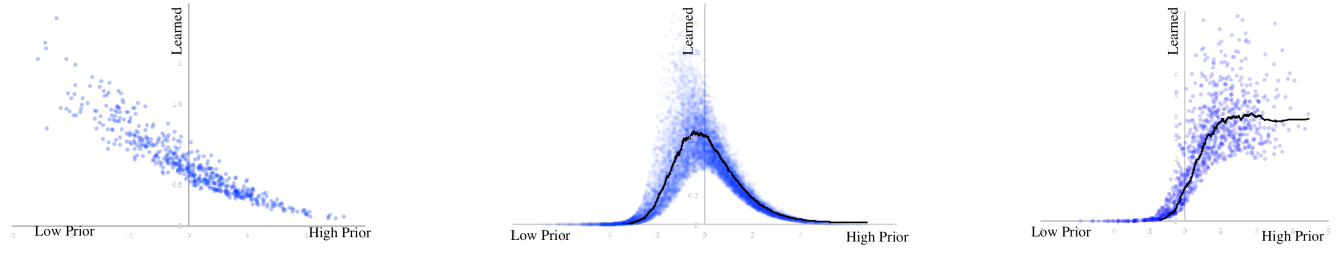


Figure 2: The same population on three different tasks. Left: a task on which beginner students learn more. Middle: a task on which medium students learn more. Right: top students learn more. Points are simulations pre-vs-learn of different students.

If we revisit our simulations and consider “pre vs learn-delta” as opposed to “pre vs post” we can gain insight into the functional form. Here we define learn-delta to be an individuals improvement from post to pre on the learning-task ($\beta_j - \alpha_j$). Figure 2 shows the “pre vs learn-delta” for three different tasks, produced by the simulation, one where beginners learn more, one where medium students learn more, and one where students with advanced prior knowledge learn more.

The graph in the middle (medium students learn more) resembles a “Gaussian” bump, whereas the graphs on the left and right look like exponential decay and growth, respectively. However, upon further inspection, we note that all of the graphs can be represented by an equation with a Gaussian bump. The exponential graphs could be considered to be left and right legs of a bump.

In order to build a functional form that matches all three scenarios (while appreciating that the “beginners learn more is much more common”) we propose a simple parametric form which can describe all three, the learning-gain function family:

Learning-Gain-Bump Family: The function of a student i from population j with learning ability θ , learning on task k with parameters ϕ is:

$$f_{\theta,\phi}(\alpha) = \alpha + \theta \cdot e^{-\frac{(\alpha-\phi_1)^2}{\phi_2}} = \beta \quad (1)$$

Where:

- α is prior ability of student i on task k
- β is posterior ability of student i on task k
- θ is “ability to learn” of population j
- ϕ is a vector of two task k specific constants.

In this model larger values of learning-ability (θ) scale up the Gaussian shaped bump.

We note that in practice most learning experiences tend to have the property that “beginners learn more” and as such an exponential decay function should often work well in practice. As such we also consider the Learning-Gain-Decay Family: $f_{\theta,\phi}(\alpha) = \alpha + (\alpha + \phi_1)^{-1} + \phi_2$

Inference is performed using a PyTorch implementation of the model, and Adam optimization to minimize the Mean Squared Error in predicting posterior (β) abilities.

4. EVALUATION

While the Learning-Gain function family seems reasonable as a hypothesis. In order to test its utility as a basis for item response theory on learning-ability, we evaluate on both simulated data with known learning-abilities and real-world data.

4.1 Simulated Evaluation

To evaluate we generated two tasks, and for each task simulated 2000 students from eight countries with a range of parameters: most importantly a single parameter which represented the latent ability to learn of a student from that population.

To evaluate, we build an inference algorithm to take the observed data produced by the simulations (the pre/post abilities of each student) and attempt to infer single value θ_j for each population j using the generative model in Equation 1. Recall that the simulations are **not** generated from our assumed function, rather it is a product of a zone-of-proximal development rather-complex simulation.

The Bayesian model, which estimates learning-ability via learning-gain-functions, is able to perfectly back-out “population ability to learn” from such simulated data (For both tasks with eight countries, $R^2 > 0.99$). In contrast a linear function was not able to fit the data nearly as well. For the task that was good for beginners it performed reasonable ($R^2 = 0.92$) whereas for the task that was good for medium prior knowledge the model was predictably unable to fit the data ($R^2 = 0.81$). While this is impressive result especially considering the complexity of the simulation, in order to consider this model useful we would like it to be able to make predictions on real-world data.

4.2 Evaluation on Real-World Data

We trained the Learning-Gain-Fn model on two real-world datasets: NWEA and ECDL. The NWEA dataset contains 69612 students from 330 schools in Grade 7 whose reading level was assessed twice (pre test and post test) using item-response theory, once in Winter and again in Spring 2017. The ECDL dataset contains data from 379 undergraduate students at the University of Alcalá (Spain) [4]. Scores for each student include four pretests and four posttests corresponding to distinct learning modules.

We compared the Learning-Gain-Fn model to a number of other plausible models: a linear model, a second-order polynomial, an exponential-decay model, and a linear model

Table 1: Results on Real-World Data

Model	Parameters per Population	Formula	NWEA Test-Set MSE	ECDL Test-Set MSE
Linear	1	$\Delta_i = \alpha_i \phi_1 + \theta_j$	56.9	0.45
Polynomial	1	$\Delta_i = \phi_1 \alpha_i^2 + \phi_2 \alpha_i + \theta_j$	56.0	0.47
Learning-Gain-Decay	1	$\Delta_i = \theta_j (\alpha_i + \phi_1)^{-1} + \phi_2$	54.9	0.44
Learning-Gain-Bump	1	$\Delta_i = \theta_j \cdot e^{-\frac{(\alpha_i - \phi_1)^2}{\phi_2}}$	53.1	0.44
Linear Multi-Theta	2	$\Delta_i = \alpha_i \theta_{j_1} + \theta_{j_2}$	55.1	0.44

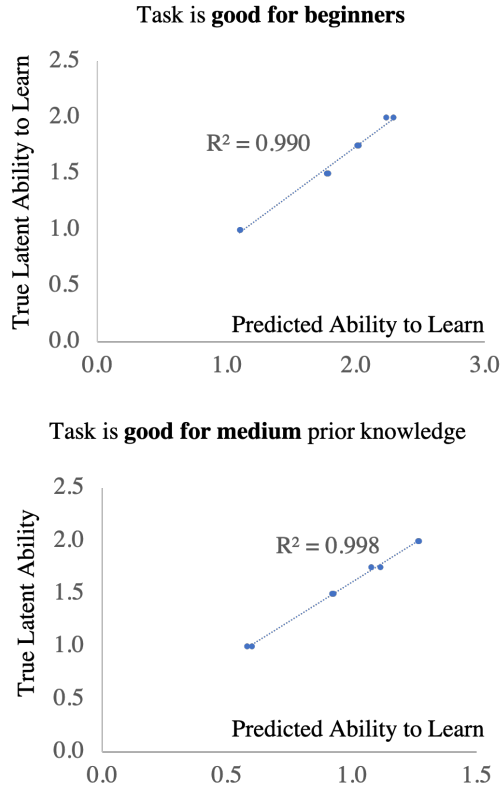


Figure 3: The simple model proposed by the Learning Gain Bump Family allows for very accurate prediction of latent ability to learn from the complex simulation.

with two parameters per population. (See Table 1 for model details.) Our primary goal was to identify a model that could best capture “ability to learn” in a *single parameter* across a variety of populations and testing scenarios. Estimation with a single parameter is important because it is more-easily interpreted—a higher value corresponds to a higher ability to learn. Each of the models we evaluated estimates “ability to learn” with a single parameter where higher values correspond to better learning ability. The exception to this rule is the Linear Multi-Theta model, which estimates ability to learn using two parameters. (See Related Work for an explanation of why this model was included.)

To compare models, we held out 10% of the data from each dataset and computed the mean-squared error when different

models made predictions about the missing data.

Notably, the Learning-Gain-Bump model outperforms all models on predicting held-out data, including the Linear Multi-Theta model. Full results are reported in Table 1. This suggests that “ability to learn” in these two cases followed a parametric form best explained by a more nuanced learning-gain function. While the gains in MSE are modest, we hypothesize that for some datasets, especially ones where the learning tasks most benefit medium strength students, the linear model will break down. We also note that the Learning-Gain-Decay and the Learning-Gain-Bump function performed very similarly – which indicates that all the tasks in this data were ones where ones where beginners learned the most.

Figure 4 shows the shape of the learning-gain-fn for different grade levels in the NWEA dataset between Winter and Spring. For every one of the 330 schools in the dataset we can now compute the ability-to-learn (θ) of the students in their population. We note that, as shown in Figure 4(b), the distribution of θ s appears to be Gaussian. Figure 4(a) also includes the learning-gain-fn for two of the top schools in the NWEA dataset. We note that it is impressive how much of ability-to-learn can be explained by which school a student went to. In the top schools (by learning-ability) students with low, medium and high prior ability substantially improve between the pre and post test.

These results are preliminary. The robust model of ability-to-learn presented in this paper will open up deeper analysis into learning in a wide range of contexts: from short tests of learning ability to evaluations of ability-to-learn in schools.

5. LIMITATIONS

Ability to learn is unlikely to be a single parameter: It is highly unlikely that a student’s ability to learn can be captured in a single number. However, this simplifying assumption proves to be convenient and useful. Often, the amount of data available to estimate parameters is small, making a model with few parameters attractive. Additionally, estimating ability to learn with a single parameter results in a model that is maximally-interpretable—the higher the number, the better the ability to learn.

There is a three-month gap between testing periods in the NWEA data: Our hypotheses about student ability to learn are based on a simulation of student learning that occurs over the course of a day. In testing these hypotheses we relied on real-world datasets that measured learning over

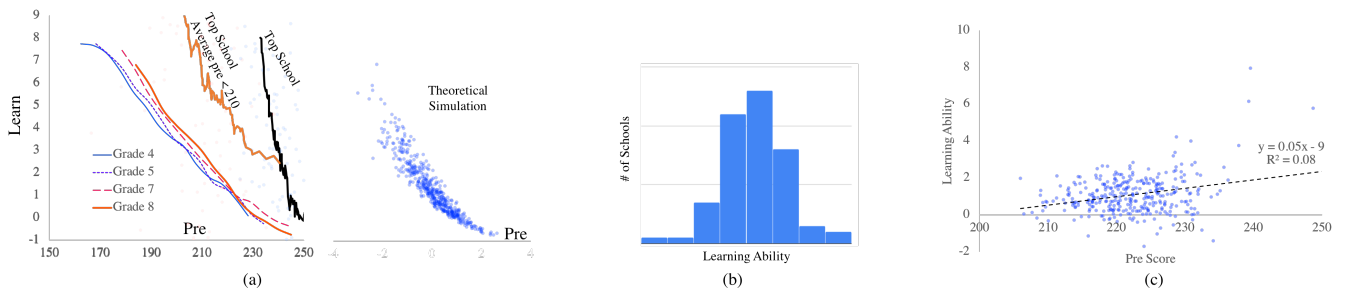


Figure 4: (a) NWEA pre-vs-learn graphs for different grade levels. Note that the distribution matches the theoretical simulated learning-task. The graph includes the pre-vs-learn for the school with the highest θ and the school with the highest θ for students with low prior ability. (b) shows the full histogram of learning abilities for different schools. (c) shows the relationship between student abilities and ability-to-learn

significantly-longer time period. For example, the two tests in the NWEA dataset that we used to measure ability to learn occurred approximately three months apart. Despite this fact, our model still fit the real-world data better than any alternative model, providing a measure of reassurance.

6. DISCUSSION

Modelling individual students’ ability to learn: The models in this paper estimate ability to learn at the group level. However, there are many cases where estimating individual students’ ability to learn would be useful as well. Due to the small number of datapoints per student, this could prove challenging. However, a hierarchical model that assumed individual students’ abilities to learn were governed by a strong group-level prior could overcome this problem.

Incorporating Pre/Post Tests: Given the function, we can incorporate this ability to learn into the traditional IRT process before and after. Specifically, the probability that a student i gets an item k right on the pre test should be, under the IRT-2PL: $p_{ik} = \sigma(\alpha_i - d_k)$ where d_k is the difficulty of item k and α_i is the same α_i that we used in our learning model. σ is the sigmoid function. Similarly the probability that student i gets a item k correct on the post test would be: $p_{ik} = \sigma(\beta_i - d_k)$ where β_i is the posterior ability of the student after the learning task. In the case where pre-post tests are real valued, we can use the logit-normal IRT proposed by Arthurs et al [2].

Fairness and Mixture Models: A Bayesian model of learning-gain-functions can do much more than simply infer ability-to-learn from pre-post tests. It would also allow for researchers to disentangle mixture distributions. This would allow researchers to identify sub-population effects within a larger population. Similarly, a robust model of ability-to-learn can be the basis of ensuring that a learning-task, and/or an education system is fair to different demographics.

Learning The Learning-Gain Function: In this paper we have modeled ability to learn as a parameter in a family of learning functions. This family of functions is Gaussian-like, a choice that was informed by observing the outcomes of a theoretically-grounded simulation. While this choice proved to have the lowest error, it is likely that another choice could offer improvements. Rather than trying a number of models,

each with its own assumptions, an alternative approach would be to use a small neural network to learn the model directly from the data.

Neural networks are universal function approximators, which means a small neural network should be able to learn the function family that serves as the best model that incorporates θ , ϕ , and α . Fears that neural networks are black-box algorithms that lack interpretability do not apply in this case—since the number of parameters is small, the learned function can be visualized directly across all values of the parameters. This approach would combine the flexibility of neural networks with the transparency and interpretability of the current models.

7. CONCLUSION

“Learning how to learn” is considered an essential skill for the 21st century [23]. Given the rapid pace of technological development, this is one of the most valuable skills an educational system can provide for its students. In recognition of this fact, the PISA 2024 test will contain an experimental section that has been explicitly designed to measure students’ ability to learn. However, few assessments have been explicitly designed to gauge this ability, meaning that the community lacks models that are capable of directly estimating this skill. In this paper we introduce a model that estimates student ability to learn using a single parameter. This model is more accurate at estimating student change in knowledge than other competing single- and multi-parameter models on two real-world datasets. Additionally, it is able to perfectly recover “ability to learn” from a complex, theoretically-grounded simulation of student learning over time. We present this work to demonstrate the value in explicitly modeling this skill, and we propose this model as a first step towards a more complete theory of understanding ability to learn.

8. REFERENCES

- [1] E. M. Anderman, B. Gimbert, A. A. O’Connell, and L. Riegel. Approaches to academic growth assessment. *British Journal of Educational Psychology*, 85(2):138–153, 2015.
- [2] N. Arthurs, B. Stenhaus, S. Karayev, and C. Piech. Grades are not normal: Improving exam score models using the logit-normal distribution. In M. C. Desmarais,

- C. F. Lynch, A. Merceron, and R. Nkambou, editors, *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada, July 2-5, 2019*. International Educational Data Mining Society (IEDMS), 2019.
- [3] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.
 - [4] L. de Marcos, E. García-López, and A. García-Cabot. Dataset on the learning performance of ecdl digital skills of undergraduate students for comparing educational gaming, gamification and social networking. *Data in brief*, 11:155–158, 2017.
 - [5] R. Glaser and A. J. Nitko. Measurement in learning and instruction. 1970.
 - [6] L. J. Hendrix, M. W. Carter, and J. L. Hintze. A comparison of five statistical methods for analyzing pretest-posttest designs. *The Journal of Experimental Education*, 47(2):96–102, 1978.
 - [7] B. Jovanovic and Y. Nyarko. A bayesian learning model fitted to a variety of empirical learning curves. *Brookings Papers on Economic Activity. Microeconomics*, 1995:247–305, 1995.
 - [8] N. M. Laird and T. A. Louis. Empirical bayes ranking methods. *Journal of Educational Statistics*, 14(1):29–46, 1989.
 - [9] Y.-J. Lee, D. J. Palazzo, R. Warnakulasooriya, and D. E. Pritchard. Measuring student learning with item response theory. *Physical Review Special Topics-Physics Education Research*, 4(1):010102, 2008.
 - [10] D. F. McCaffrey, J. Lockwood, D. M. Koretz, and L. S. Hamilton. *Evaluating Value-Added Models for Teacher Accountability. Monograph*. ERIC, 2003.
 - [11] M. S. McCall, C. Hauser, J. Cronin, G. G. Kingsbury, and R. Houser. Achievement gaps: An examination of differences in student achievement and growth. the full report. *Northwest Evaluation Association*, 2006.
 - [12] M. S. McCall, G. G. Kingsbury, and A. Olson. Individual growth and school success. a technical report from the nwea growth research database. *Northwest Evaluation Association*, 2004.
 - [13] J. P. Meyer and S. Zhu. Fair and equitable measurement of student learning in moocs: An introduction to item response theory, scale linking, and score equating. *Research & Practice in Assessment*, 8:26–39, 2013.
 - [14] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012.
 - [15] M. Ramscar, P. Hendrix, C. Shaoul, P. Milin, and H. Baayen. The myth of cognitive decline: Non-linear dynamics of lifelong learning. *Topics in cognitive science*, 6(1):5–42, 2014.
 - [16] D. D. Ready. Associations between student achievement and student learning: Implications for value-added school accountability models. *Educational Policy*, 27(1):92–120, 2013.
 - [17] D. L. Schwartz, J. D. Bransford, D. Sears, et al. Efficiency and innovation in transfer. pages 1–51.
 - [18] D. L. Schwartz, R. Lindgren, and S. Lewis. Constructivism in an age of non-constructivist assessments. pages 34–61.
 - [19] D. L. Schwartz and T. Martin. Inventing to Prepare for Future Learning: The Hidden Efficiency of Encouraging Original Student Production in Statistics Instruction. 22(2):129–184.
 - [20] A. M. Thomson and C. Lamy. Functional maps of neocortical local circuitry. *Frontiers in neuroscience*, 1:2, 2007.
 - [21] Y. M. Thum and C. H. Hauser. Nwea 2015 map norms for student and school achievement status and growth. *Portland, OR: NWEA*, 2015.
 - [22] E. Weber. Quantifying student learning: how to analyze assessment data. *The Bulletin of the Ecological Society of America*, 90(4):501–511, 2009.
 - [23] C. E. Weinstein. Learning how to learn: An essential skill for the 21st century. *Educational Record*, 66(4):49–52, 1996.

Iterative Feature Engineering Through Text Replays of Model Errors

Stefan Slater
University of Pennsylvania
slater.research@gmail.com

Ryan S. Baker
University of Pennsylvania
rybaker@upenn.edu

Yeyu Wang
University of Wisconsin – Madison
wangyeyu215@gmail.com

ABSTRACT

Feature engineering, the construction of contextual and relevant features from system log data, is a crucial component of developing robust and interpretable models in educational data mining contexts. The practice of feature engineering depends on domain experts and system developers working in tandem in order to creatively identify actions and behaviors of interest. In this paper we outline a method of iterative feature engineering using the misclassifications of earlier models. By selecting cases where earlier models and ground truth disagree, we can focus attention on specific behaviors, or patterns of behavior, that a model is not using in its predictions. We show that iterative feature engineering on cases of false positives and false negatives improved a model predicting quitting in an educational video game by 15%. We close by discussing applications of this method for addressing model performance gaps across different classes of learners, as well as precautions against model overfitting with using this method of feature engineering.

Keywords

Feature engineering, knowledge engineering, games, text replays

1. INTRODUCTION

Educational games and digital simulations are powerful educational tools that have seen increasing use in classrooms within the last decade. These digital environments afford students rich opportunities to engage deeply with content, adopt new and different identities [6], explore personally relevant domains [8], and develop non-cognitive skills such as productive persistence [17]. The adoption of educational games as tools for learning has been accompanied with an increasing focus on educational games as a medium for the application of educational data mining. The medium of educational games presents challenges for EDM methodologies, however, as the relative complexity of student behaviors in games can be quite broad when compared to more constrained environments such as intelligent tutoring systems (ITS).

Given the more complex behaviors possible for students in these environments, researchers studying learning in digital environments and games are able to identify and predict more

complicated cognitive and non-cognitive constructs. Some examples of constructs identified in games include persistence [14], elegant problem solving [13], seriousness [5], carefulness [4], computational thinking [1], and mental demand [31].

This increased complexity places an increased importance on the feature engineering and/or knowledge engineering steps of the data science pipeline. Expert knowledge is often crucial for understanding specific patterns of behavior within educational games and simulations. For example, deep understanding of both gameplay design and conceptual understanding of physics were needed to develop a model of whether students had implicit conceptual understanding of physics based on how they responded to balls of different colors (connoting mass) in a physics game [22]. This understanding has driven feature engineering in many of these cases. Previous work by [23] has shown that feature selection and feature engineering of variables with high construct validity can lead to better model performance on unseen data. The question, then, is how we as researchers can quickly and effectively identify the specific patterns of player behavior that “matter” – how can we best separate the signal from noise in a large, complex dataset on student behavior and interaction?

Historically, social sciences researchers have addressed the complexity of human behaviors by combining qualitative methods providing “thick description” of actions [7] with quantitative methods to make scalable and general claims. However, the considerable amount of behavioral log data generated by modern learning systems poses a challenge to the qualitative analysis of human behaviors. One approach, termed “closing the interpretive loop” [24], is to refine and validate a model by looping back to the raw data, and checking whether the model and data are consistent. In an application of this method, [12] constructed a model to investigate how interactive indicators in the *Jaune Fluo* dataset relate to emotions in learning. By returning to and leveraging raw transcription data, they gained insights about micro-level interactions between speakers that could be used to drive modeling.

In this paper we propose a related approach -- a method for selecting specific cases of relevance from a larger dataset for further analysis, using instances of model mis-prediction. By adopting an iterative approach to model selection and feature engineering, we can use cases of false positives and false negatives to identify the specific cases where the model fails to accurately match student data, to better uncover relevant gameplay behaviors and patterns. We can then employ qualitative techniques to these cases to better understand what is occurring, and use these findings for additional feature engineering and model iteration. By closing the interpretive loop, we not only gain deeper understanding of the data, but also generate new contextual features for modeling in a way that is closely tied to observed patterns of behaviors in the data. We apply this method in the

Stefan Slater, Ryan Baker and Yeyu Wang "Iterative Feature Engineering Through Text Replays of Model Error" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 503 - 508

broader context of studying student quitting behavior in the educational physics simulation game *Physics Playground* [28].

2. METHODS

Data for this work comes from a series of randomized controlled trials (RCTs) conducted at middle schools in Pennsylvania and Florida during the spring of 2019 using the educational physics simulation game *Physics Playground*, courtesy of the *Physics Playground* team.

Physics Playground teaches elementary physics concepts such as conservation of momentum and torque through a sandbox environment where players are tasked with drawing simple machines that move a ball to a balloon elsewhere in the level. Students receive badges for successfully solving levels, and are able to use these badges to unlock different types of music, custom balls, and other cosmetic changes within the game. *Physics Playground* also contains in-game hints and scaffolds, accessible through a help button on the UI.

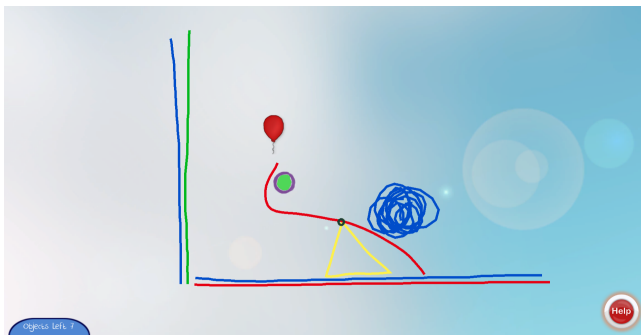


Figure 1. Physics Playground. The author has (unsuccessfully) built a lever and dropped a weight on it in an attempt to move the ball to the balloon. The help button (bottom right) and object counter (bottom left) are also pictured. The author would like to note that they are not a physicist.

A total of 96 students participated in the study. The RCTs were designed to test the effectiveness of several types of learning supports for *Physics Playground* on learning gains in the game. Students spent a total of ~110 minutes of class time playing *Physics Playground* in between a physics knowledge pretest and posttest, across four days. In the treatment condition, students were able to access a help button in the game UI that allowed the student to select multiple types of scaffolds to watch. Through the help button, students were able to receive help related to the use of game tools and mechanics, worked example solutions, and abstracted physics concepts. Students in the control condition were automatically prompted to use this button after three minutes had elapsed, but were unable to access the help button before that point. Preliminary analyses identified no significant differences in posttest scores or learning gains between conditions, so for the current study we combined these two groups and ignored condition assignment. Additional details on the study and its overall findings can be found in [29].

2.1 Data Structuring, Preprocessing, and Labeling

Gameplay data from the study were collected by the game's servers and output as .json files. A total of 703,765 records of student gameplay were collected during the study, where one record is a single logged student action in the game.

Several pre-processing steps were taken to prepare the data for analysis. Three students who did not complete the consenting process for the study were removed from the dataset. Events which occurred outside of study hours were also removed from the dataset. These events were due to students continuing to play the game in their free time. Attempts which were shorter than two seconds were also removed from the dataset. These attempts often consisted of students rapidly pressing the spacebar to reset their current level, without taking any in-game actions.

We also added additional contextual information into the dataset. We added the *Physics Playground* q-matrix into the dataset, which consists of the mapping between levels and physics constructs to be taught, as well as the simple machines associated with each level's solution. We added a series of session, visit, and attempt IDs to each record. A "session" is a length of time from student login to student logout. New sessions can begin when a student begins playing *Physics Playground* for the first time each day, or, when a student refreshes their browser. A total of 586 gameplay sessions were recorded, for an average of six sessions for each student. It is worth noting that students played the game for four days within the study; the higher average number of sessions is because students could accidentally refresh their browsers, or hit the "back" button, which began the logging of a new session. Within each session are "visits" – a visit lasts from the beginning of a level to the end of a level, whether the student solves that level successfully or quits to go to a different level. We identified 2906 total visits, with an average of 30 visits for each user – slightly less than the 34 levels available to play in the game for the current study. Finally, within each visit are "attempts" – an attempt begins any time that the level is initialized, and ends when a student either successfully solves the level, restarts the level, or quits the level. We identified 16,546 total attempts in the game, with an average of 172 for each user.

Given this structure of sessions, visits, and attempts, we defined a "quit event" as any time a student begins a new visit, within the same session, when their previous attempt was not successful. This represents a student failing to solve a level, leaving that level entirely, and playing a different level within the game. From each quit event, we labeled each record that happened up to 120 seconds before the event as "quit", and all other records as "not quit". Previous work on predicting quitting in *Physics Playground* used aggregations of 60-second clips within each attempt, e.g. [10]. In contrast, our method of labeling quitting at the event level, and up to 120 seconds prior, allows us to identify quitting across attempts, and sometimes across visits, in order to allow earlier detection and intervention by automated systems or in-classroom educators.

2.2 Initial Feature Engineering and Model Fitting

Drawing on previous literature that has explored *Physics Playground* [10,13], we developed an initial set of 32 features to use in predicting student quit behavior. These features included counts of each type of object or simple machine (weights, ramps, levers, pendulums, springboards, freeforms, and pins) that the student had drawn total and per attempt, the number of times students went some number of seconds without recording an action (5, 10, 15, 30, and 60 seconds), and whether students used each type of scaffold (worked examples, game tools, and physics animations) as well as the number of scaffolds that they used total and in each attempt. We also developed features to capture the amount of time that students spent using scaffolds, as well as the

amount of time that had passed (in seconds) since the last time a scaffold was used. Finally, we recorded the elapsed time of each attempt, as well as the total elapsed time of the session, and the number of badges that students had earned so far.

For modeling quit behavior, we chose to use a relatively simple logistic regression model rather than more sophisticated algorithms such as a decision tree, gradient classifier, or recurrent neural network. Regression-based models are easier to implement into *Physics Playground's* Unity-based architecture than more sophisticated machine learning models. We used five-fold student-level cross validation in RapidMiner 9.4 [15]. We did not use any feature selection procedures for modeling; each feature was used as a component in the final model. We did not believe that our feature space was large enough to warrant feature selection. We used AUC ROC as our goodness metric, as we were more interested in overall model performance than optimizing our quit prediction threshold.

This initial model, which we will call the “original” model with “original” features in this paper, has an AUC of 0.688.

2.3 Error Identification and Feature Re-Engineering

Using the confusion matrix of this initial model, we identified all false positives and false negatives and mapped these events onto the attempts in the dataset. In other words, if *any* record within an attempt contained a case of model mis-prediction, we labeled the entire attempt as a mis-prediction. This resulted in 1,487 attempts labeled as cases of false negatives (9% of all attempts) and 298 cases of false positives (2% of all attempts). We then used text replays [2] to qualitatively code these attempts for patterns of engagement or behavior that we believed could be related to quitting behavior in players. Text replays have been used previously to conduct in-depth study of other constructs such as gaming the system [19], as well as to obtain training labels for the development of detectors [21, 23, 5]. [19]’s research shows that they can be a powerful tool for developing thick descriptions of learner behavior, and that this deeper understanding can lead to substantially better models of that behavior [18]. We randomly selected 100 examples each of false positives and false negatives for this coding process and conducted text replays on these attempts, taking notes on potential new features which could capture behaviors that we observed in the data. This coding procedure was done by a single researcher. As in [19], reliability measures were not obtained, as the goal was to develop new features that could be applied to the data programmatically rather than to develop a scalable human-based coding method. In our coding, we also viewed only single attempts, not looking at preceding or subsequent attempts (as in most prior uses of text replays).

Overfitting is an inherent concern for iterative feature engineering processes; we will discuss in the discussion section why overfitting may be particularly concerning for this paper’s method. Because we wanted to overfit as little as possible, we only looked at text replays of false negatives and false positives. We intentionally did not view text replays of cases of true positives or true negatives. In other words, when we saw a behavioral pattern in false positives or false negatives, we did not double-check whether it was also seen in true positive or true negative cases, with a goal of deriving more features rather than attempting to conduct feature selection by hand by looking at the data (which could increase risk of over-fitting).

Our text replay and qualitative coding processes identified 14 additional features that we then developed software to apply to the dataset. Four of these new features related to scaffold use: **Multiple Uses Of Same Scaffold**, the number of times a student used the same learning support more than once in the same attempt; **Short Scaffold Time**, the number of times a student spent less than five seconds interacting with a scaffold; **Early Scaffold Use**, the number of times that a scaffold use appeared in the first third of actions that a student took in a given attempt; and **Multiple Scaffolds In Attempt**, the number of times that a student used more than one scaffold in the same attempt. Four features related to attempt duration: **Long Attempt Count**, the total number of attempts over three minutes; **Average Last Three Attempt Times**, the average duration of the last three attempts that a student had; **Attempt Time Standard Deviation**, the standard deviation of time across all student attempts so far; and **Previous Attempt Duration**, the duration of the attempt immediately before the current one. Three new features related to machine drawing and use: **Net Objects Drawn**, the number of objects a student drew on the current attempt minus the number of objects a student erased; **Time Spent Drawing**, the total elapsed time between the start and end of a student drawing a machine; and **Unexpected Machine Used**, whether a student drew a machine that was not associated with the knowledge component of the current level. We also created a feature for **Consecutive Nudges**, the number of consecutive times the student clicked on the ball to attempt to move it (cf. [9]), and a feature for **Recently Restarted**, whether the student restarted an attempt within the last 120 seconds. A restart is when a student unsuccessfully solves a level, but retries the same level rather than quitting and going to a new one.

The final re-design to our model, which we called **Quit Flush**, went beyond just creating a new feature. During coding, especially for false positives, we noticed that the model would continue to predict quitting after a quit event when the student did not subsequently quit. A student would begin a new attempt with the model already predicting that the student would quit. Then, some amount of time after the attempt had started, the quit prediction would drop off, and the student would go on to either restart the level or complete it successfully. We hypothesized that this was because the student may have quit in an earlier attempt, and the model had not yet caught up to the student’s new behavioral patterns in a different visit. Therefore, we constructed a separate dataset, which we called the Quit Flush dataset. In this dataset, we reset the values of *all* features following a quit event, starting the model over again from a blank slate whenever a quit was identified.

Following this feature engineering process, we replicated the model fitting steps of the original model exactly. We also fit a series of models where we held out each new feature, to examine the performance gain from adding each feature into the new model.

3. RESULTS

3.1 Original Model vs. Enhanced Model

Our enhanced model, using all 14 newly engineered features (but not including the quit flush), produced an AUC of 0.812 – a gain of almost 0.10, and a 15% improvement over the original model. The enhanced model’s performance is comparable to the best performing models developed by [10], even with the limitation of a relatively simple logistic regression rather than the more sophisticated classification algorithm used in that paper. We will

call this model the “enhanced model” with “enhanced features” in this paper.

3.2 Enhanced Model vs. Quit Flush Model

The quit flush model, using all 14 newly engineered features and resetting all features’ values after a quit, produced an AUC of 0.616 – slightly worse than even the original model. The poor performance of the quit flush model suggests that student quitting events are “sticky”, and that moving between levels does not necessarily indicate that a student starts working productively.

An ROC curve comparison between the AUC of the original model, enhanced model, and quit flush model is given in Figure 2.

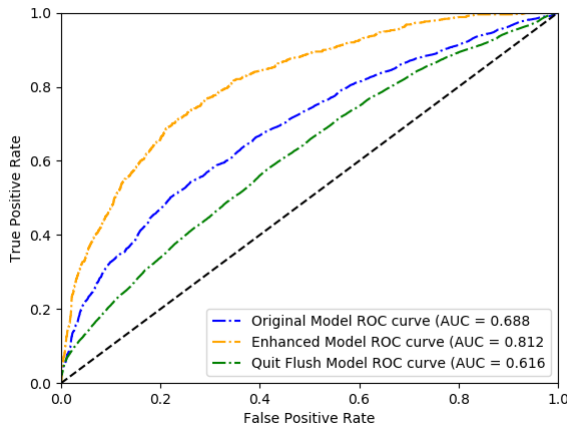


Figure 2. Comparison between Original Model AUC, Enhanced Model AUC, and Quit Flush Model AUC.

3.3 Leave One Out Model Results

Following our fitting of the enhanced model and the quit flush model, we then fit subsequent iterations of the enhanced model, holding one feature out each time. The purpose of this analysis was to identify the contribution that each individual feature made to the overall performance of the model.

Table 1. Comparison of feature impact on enhanced model performance.

Feature Name	AUC When Held Out	Delta
Consecutive Nudges	0.742	-0.070
Early Scaffold Used	0.793	-0.019
Previous Attempt Duration	0.812	--
Short Scaffold Use	0.812	--
Attempt Longer Than 3m	0.812	--
Average Last 3 Attempts	0.812	--
Uses of Same Scaffold	0.812	--
Multiple Scaffold Uses	0.812	--
Recent Restart	0.812	--
StD Attempt Duration	0.812	--
Time Drawing Objects	0.812	--

Unusual Object Used	0.812	--
Net Objects Drawn	0.812	--

We found that the performance increase of the enhanced model was driven primarily by just two features – **consecutive nudges**, and **early learning support use**. Student use of consecutive nudges was also found to be associated with student quitting in research by [9] – it is possible that students using nudges repeatedly could indicate that the student is trying to make an ineffective solution work when they cannot figure out a more productive means of solving the level. Further analysis of this behavior, perhaps incorporating qualitative interview data from students during or immediately after gameplay, could contribute to a better understanding of this behavioral pattern. Early use of learning support could either indicate a student who is completely stuck and doesn’t know where to start, or a lack of willingness to put in effort to solve a level, either of which could lead a student to quit.

4. DISCUSSION AND CONCLUSION

In this paper we demonstrate that iterative feature engineering using cases of model mis-prediction, conducting qualitative coding of text replays on instances of false positives and false negatives, can enhance model performance. We found that an “enhanced” model, using features that we developed through this method, performed 15% better than an “original” model on the same dataset. This performance gain came from just two of the 14 features that we iteratively engineered. We have several potential applications for this methodology, and an important caveat to make.

4.1 Applications of Iterative Feature Engineering

Educational data mining has been applied to a wide variety of problems, and we believe that iterative feature engineering may have specific learning contexts where it is more useful. Specifically, this technique relies on having rich, nuanced log data from which specific details of student interaction with the system can be drawn. For relatively simple contexts, such as students working on an online quiz system with very few choices or different components, this method may be more difficult to apply successfully, as there may not be enough variation in behavior for iterative feature engineering to be useful. On the other hand, contexts with rich contextual data may be better suited for this method. This method may be particularly useful for improving prediction models that were already developed using text replays [20, 11, 5, 23], though there is not a principled reason why the method could not be useful even for models initially developed using other methods. In applying iterative feature engineering to models not developed using text replays, it may be relevant to consider whether the behavior can be identified and understood from text replays – some forms of affect, for instance, may be difficult for humans to identify directly from this type of data.

4.2 Using Iterative Feature Engineering to Address Uneven Quality Across Populations

In this work, we applied our iterative feature engineering process to the entire dataset. However, recent papers have found that many EDM models can perform unevenly for different populations of learners, such as rural students versus urban students or non-native speakers versus native speakers (see review

in [3]). We believe that the approach proposed in this paper may be a useful tool for fixing this type of inequity. Training separate models on each sub-population within the full dataset, and conducting qualitative coding on instances of error within these sub-population, could highlight different behavioral patterns seen for individuals in different groups. Population-specific features could then be engineered to equalize the performance of the model across groups of learners.

4.3 Caveat – The Potential for Overfitting

Consider this technique taken to its logical conclusion – monkeys on infinite typewriters, endlessly thresholding and re-thresholding features, defining and re-defining cutoffs, until the billionth permutation of this process produces a model with no error. This model, obviously, would be massively overfit and of next to no use in any broader context, such as on a new class of students. In practice, there are likely not enough educational data monkeys in the world to produce a model with no error. That said, conducting several cycles of iteration on the same dataset does run the risk of overfitting one's feature engineering process, and subsequently the model, to the particulars of the dataset being used. Therefore, this is likely a method best used only limited times over the course of model development. Ultimately, fully-withheld test datasets – or better yet, the collection of new datasets after the fitting process is complete -- should be used in final evaluation of a model (as seen in the trajectory of gaming the system modeling between [19] and [18]). It is not yet known how much iteration of this nature is beneficial, before diminishing returns or overfitting occur. It may be a valuable step for future research to investigate iterating multiple times and observing changes in model performance, identifying the elbow point for improvement. From the perspective of quantitative ethnography, researchers might consider stopping the iterative process when reaching *theoretical saturation*, seeing more data but failing to generate new insights [25].

4.4 Alternative Approaches to Feature Engineering and Text Replays

In this paper, we started with a model with initial features and then refined the model by examining the misclassified cases and deriving new features based on the qualitative interpretations of game play behaviors. However, this is not the only way that an iterative process of feature engineering could be conducted. [26] outline an alternative approach for constructing a theoretical-based and analytics-driven model by grounding analyses in qualitative data and exploring the pattern of data before model construction. Epistemic Network Analysis (ENA) models the co-occurrence of behaviors based on coded qualitative data and unpacks the complexity in the learning process [27]. Used first in epistemic games, other scholars have begun applying ENA techniques to this same problem. For example, [9] adopted ENA to explore why learners quit levels in *Physics Playground*, investigating cognitive processes based on student interaction with the game. Their study identified that students who crystalize their problem-solving strategy at the beginning of gameplay are more likely to quit the levels. This behavior pattern suggests new features to be engineered for the future study of quit model prediction.

4.5 Further Model Development and Future Directions

Poor performance of the quit flush model suggests that a student-level model may be beneficial to predicting quitting. Originally,

our justification for creating the quit flush model was that we observed cases of quitting being predicted at the beginning of an attempt, and we hypothesized that this could be due to the model continuing to predict quitting behavior immediately after a quit event occurs. We anticipated that a quit flush feature would improve overall model performance by addressing these cases; however, the quit flush model performed significantly worse than both the enhanced and original models. Given this difference in performance, it is possible that an enhanced model which uses features aggregated across visits or even sessions of play, could be a more effective predictor than the simple count and duration features that we used in the current work. Previous work on *Physics Playground* has identified the existence of player typologies [30], representing distinct approaches that groups of players employ when playing *Physics Playground*. This work found that students who played *Physics Playground* could be assigned to one of three classes: *achievers*, motivated by in-game rewards such as badges; *explorers*, motivated by the ability to explore the game space, design interesting and unique machines, and push the boundaries of the physics simulation, and *disengaged* players, those players who did not engage with the game to the same degree as their peers. These classes of players could serve as valuable student-level features that inform overall patterns of play. Further work on this topic may also use features aggregated up to each level, as previous work has [10, 13]. While we did not use these features for the current work, because of difficulties in generating these variables at run-time inside the game environment, future work that is focused on using the quit model for analysis entirely outside of the game might benefit from using these various levels of aggregation.

This work has leveraged qualitative analysis in a somewhat different fashion than prior efforts within the EDM community. Qualitative techniques are not new within the fields of educational data mining and learning analytics. Text replays – human review of student behavior to generate labels – have been used for over a decade [2]. Similar work, such as [16] has focused on generating human-readable samples of student-tutor interactions. However, these methods are usually used to generate ground truth labels in order to construct models, or to better understand the relationships that these behaviors have with one another. In this paper, we utilize qualitative coding of student behaviors to develop features that subsequent models can be trained on. By bringing more qualitative understanding and analysis into educational data mining and learning analytics, and synthesizing these approaches with quantitative modeling practices, we can develop models that perform better and are more understandable by the research community.

5. ACKNOWLEDGMENTS

We would like to thank the *Physics Playground* team for their assistance in producing the datasets used.

6. REFERENCES

- [1] Almeda, M.V., Rowe, E., Asbell-Clarke, J., Scruggs, R., Baker, R., Bardar, E., Gasca, S. (2019) Modeling Implicit Computational Thinking in Zoombinis Mudball Wall Puzzle Gameplay. To appear in *Proceedings of the 2019 Technology, Mind, and Society Conference*.
- [2] Baker, R.S., & de Carvalho, A.M.J.A. (2008). Labeling Student Behavior Faster and More Precisely with Text Replays. *Proc. of the 1st Int'l Conf. on Educational Data Mining*, 38-47.

- [3] Baker, R.S., Ogan, A.E., Madaio, M., Walker, E. (2019) Culture in Computer-Based Learning Systems: Challenges and Opportunities. *Computer-Based Learning in Context*, 1(1), 1-13.
- [4] Banawan, M.P., Rodrigo, M.M.T., & Andres, J.M.A.L. (2015). An Investigation of Frustration Among Students Using Physics Playground. *Proceedings of the 23rd International Conference on Computers in Education*.
- [5] DiCerbo, K., & Kidwai, K. (2013). Detecting player goals from game log files. *Proceedings of the 6th Annual Conference on Educational Data Mining*.
- [6] Gaydos, M.J., & Devane, B.M. (2019). Designing for identity in game-based learning. *Mind, Culture, and Activity*, 26(1), 61-74.
- [7] Geertz, C. (1973). Thick description: Toward an interpretive theory of culture. In *The interpretation of cultures: Selected essays* (pp. 3–30). New York: Basic Books.
- [8] Holbert, N., & Wilensky, U. (2019). Designing educational video games to be objects-to-think-with. *Journal of the Learning Sciences*, 28(1), 32-72.
- [9] Karumbaiah, S., Baker, R.S., Barany, A., Shute, V. (2019) Using Epistemic Networks with Automated Codes to Understand Why Players Quit Levels in a Learning Game. *Proc. of the 1st Int'l Conference on Quantitative Ethnography*, 106-116.
- [10] Karumbaiah, S., Baker, R.S., & Shute, V. (2018). Predicting Quitting in Students Playing a Learning Game. *Proceedings of the 11th International Conference on Educational Data Mining*, 21-31.
- [11] Lee, D.M.C., Rodrigo, M.M.T., Baker, R.S., Sugay, J.O., & Coronel, A. (2011). Exploring the relationship between novice programmer confusion and achievement. *Proceedings of the 2011 International Conference on Affective Computing and Intelligent Interaction*, 175-184.
- [12] Lund, K., Quignard, M., & Shaffer, D.W. (2017). Gaining Insight by Transforming between Temporal Representations of Human Interaction. *Journal of Learning Analytics*, 4(3), 102-122.
- [13] Malkiewicz L.J., Baker, R.S., Shute, V., Kai, S., Paquette, L. (2016). Classifying behavior to elucidate elegant problem solving in an educational game. *Proceedings of the 9th International Conference on Educational Data Mining*, 448-453.
- [14] Malkiewicz, L.J., Lee, A., Slater, S., Xing, C., & Chase, C.C. (2016). No Lives Left: How Common Game Features Could Undermine Persistence, Challenge-Seeking and Learning to Program. *Proceedings of the 2016 International Conference of the Learning Sciences*, 186-193.
- [15] Mierswa, I., & Klinkenberg, R. (2019). RapidMiner Studio (9.4). Retrieved from <https://rapidminer.com>.
- [16] Mostow, J., Beck, J., Cen, H., Cuneo, A., Gouvea, E., & Heiner, C. (2005). An educational data mining tool to browse tutor-student interactions: Time will tell. *Proceedings of the Workshop on Educational Data Mining, National Conference on Artificial Intelligence*, 15-22.
- [17] Owen, V.E., Roy, M.H., Thai, K.P., Burnett, V., Jacobs, D., Keylor, E., & Baker, R.S. (2019). Detecting Wheel-Spinning and Productive Persistence in Educational Games. *Proc. of the 12th International Conf. on Educational Data Mining*, 378-383.
- [18] Paquette, L., Baker, R.S. (in press) Comparing machine learning to knowledge engineering for student behavior modelling: A case study in gaming the system. To appear in *Interactive Learning Environments*.
- [19] Paquette, L., de Carvalho, A.M.J.A., Baker, R.S., & Ocumpaugh, J. (2014). Reengineering the Feature Distillation Process: A Case Study in the Detection of Gaming the System. *Proc. of the 7th Int'l Conf on Educational Data Mining*, 284-287.
- [20] Richey, J.E., Andres-Bray, J.M.L., Mogessie, M., Scruggs, R., Andres, J.M.A.L., Star, R.J., Baker, R.S., McLaren, B.M. (in press). More Confusion and Frustration, Better Learning: The Impact of Erroneous Examples. To appear in *Computers and Education*.
- [21] Rodrigo, M.M.T., Baker, R.S.J.d., McLaren, B., Jayme, A., Dy, T. (2012) Development of a Workbench to Address the Educational Data Mining Bottleneck. *Proceedings of the 5th International Conference on Educational Data Mining*, 152-155.
- [22] Rowe, E., Asbell-Clarke, J., Baker, R.S., Eagle, M., Hicks, A.G., Barnes, T.M., Brown, R.A., Edwards, T. (2017) Assessing Implicit Science Learning in Digital Games. *Computers in Human Behavior*, 76C, 617-630.
- [23] Sao Pedro, M.A., Baker, R.S.J.d., Gobert, J., Montalvo, O. Nakama, A. (2013) Leveraging Machine-Learned Detectors of Systematic Inquiry Behavior to Estimate and Predict Transfer of Inquiry Skill. *User Modeling and User-Adapted Interaction*, 23 (1), 1-39.
- [24] Shaffer, D. W. (2017). Quantitative ethnography. Madison, WI: Cathcart Press.
- [25] Shaffer, D. W. (2018). Big data for thick description of deep learning. In K. Millis, D. Long, J. Magliano, and K. Weimer (Eds.), *Deep learning: Multi-disciplinary approaches* (pp. 262-275). NY, NY: Routledge.
- [26] Shaffer, D. W. & Ruis, A. R. (2017). Epistemic network analysis: A worked example of theory-based learning analytics. In C. Lang, G. Siemens, A. Wise, & D. Grasevic (Eds.), *Handbook of Learning Analytics* (pp. 175–187). Society for Learning Analytics Research.
- [27] Shaffer, D. W., Collier, W., & Ruis, A. R. (2016). A tutorial on epistemic network analysis: Analyzing the structure of connections in cognitive, social, and interaction data. *Journal of Learning Analytics*, 3(3), 9–45.
- [28] Shute, V. J., Almond, R. G., & Rahimi, S. (2019). Physics Playground (Version 1.3) [Computer software]. Tallahassee, FL: Retrieved from <https://pluto.coe.fsu.edu/ppteam/pp-links/>
- [29] Shute, V., Rahimi, S., & Smith, G. (2019). Game-Based Learning Analytics in Physics Playground. In *Data Analytics Approaches in Educational Games and Gamification Systems* (pp. 69-93). Springer, Singapore.
- [30] Slater, S., Bowers, A., Kai, S., & Shute, V. (2017). A Typology of Players in the Game Physics Playground. *Proceedings of the 2017 DiGRA International Conference*.
- [31] Wiggins, J.B., Kulkarni, M., Min, W., Mott, B., Boyer, K.E., Wiebe, E., & Lester, J. (2018). Affect-based Early Prediction of Player Mental Demand and Engagement for Educational Games. *Proceedings of the 14th Artificial Intelligence and Interactive Digital Entertainment Conference*, 243-249.

Course Recommender Systems with Statistical Confidence

Zachary Warnes
Department of Data Science
and Knowledge Engineering,
Maastricht University, Maastricht 6200MD,
The Netherlands
z.warnes@student.maastrichtuniversity.nl

Evgueni Smirnov
Department of Data Science
and Knowledge Engineering,
Maastricht University, Maastricht 6200MD,
The Netherlands
smirnov@maastrichtuniversity.nl

ABSTRACT

Selecting courses in an open-curriculum education program is a difficult task for students and academic advisors. Course recommendation systems nowadays can be used to reduce the complexity of this task. To control the recommendation error, we argue that course recommendations need to be provided together with *statistical* confidence. The latter can be used for computing a statistically valid set of recommended courses that contains courses a student is likely to take with a probability of at least $1 - \epsilon$ for a user-specified significance level ϵ . For that purpose, we introduce a generic algorithm for course recommendation based on the conformal prediction framework. The algorithm is used for implementing two conformal course recommender systems. Through experimentation, we show that these systems accurately suggest courses to students while maintaining statistically valid sets of courses recommended.

Keywords

Recommender Systems, Course Recommendation, Conformal Prediction

1. INTRODUCTION

Recommender systems are systems capable of predicting the preferences of users over sets of items [1]. They can be found almost everywhere in the digital space, shaping the choices we make, the products we buy, the books we read, or the movies we watch. The range of applications of recommender systems has been broadened recently to the education domain, especially in higher education [5]. There are systems reported that provide recommendations for academic choices, learning activities, learning resources, and learning collaborations [14].

Among the recommender systems for academic choices, there exists a particular interest in systems that recommend courses [3]. There is a wide range of such systems that differ in the underlying recommendation mechanism, accuracy, type of

recommendations (courses, course sequences, course concentrations), and type of representation. It has been recently recognized that course recommender systems need to be safe [11]; i.e., course recommendations need to be provided with confidence information that will help a student to make a better course selection. There exist different approaches to delivering such confidence information from course preference ranks estimated by the underlying recommendation mechanisms [3, 6, 10, 12] to separate warning modules [11]. The characteristic feature of these approaches is that they are heuristic, and thus they do not provide any theoretical guarantees for the quality of course recommendation.

In this paper, we argue that course recommendations need to be supported with *statistical* confidence. This confidence will allow computing a statistically valid set of recommended courses that contains courses a student is likely to take with a probability of at least $1 - \epsilon$ for a user-specified significance level ϵ . To achieve this, we employ the well-known conformal-prediction framework [4, 15, 16]. We design a generic algorithm for conformal course recommendation capable of computing statistically valid sets of courses for students. The algorithm is used for implementing two conformal course recommender systems that employ a content-based recommendation mechanism. The first system is instance-based, and the second system is an exemplar-based system [13].

The conformal course recommender systems have been implemented for academic advising of University College Maastricht, a Liberal Arts Bachelor study with an open curriculum. In this study, students personalize their program by selecting courses that align with their academic and personal interests. In total, students choose around 40 out of the 160 possible educational modules; i.e., they create a program by selecting one path out of $\binom{160}{40}$ possible. Our recommender systems are tested to facilitate this process. The initial experimental results show that the systems accurately recommend courses while providing statistically valid sets of courses recommended.

The rest of the paper is organized as follows. The related work is provided in Section 2. Section 3 formalizes the task of course recommendation. The course and student topic models used for course recommendation are briefly described in Section 4. Section 5 introduces the generic algorithm for conformal course recommendation and its instantiations: the instance-based and exemplar-based recommender sys-

Zachary Warnes and Evgueni Smirnov "Course Recommender Systems with Statistical Confidence" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 509 - 515

tems. The student-course data is described in Section 6. Section 7 provides the experiments and discussion. Finally, Section 8 concludes the paper.

2. RELATED WORK

Course recommender systems received significant attention since the very first publications [12, 18, 17]. Meanwhile, these systems have become very diverse. Following the main trends in recommender-system research there are different types of course recommender systems: content-based systems [10, 11], collaborative-filtering systems [3], hybrid systems [3, 6], and popularity-based ranking systems [6]. Most of these systems are capable of providing (explicitly/implicitly) confidence information for course recommendations. However, this does not give any guarantee for the quality of course recommendation in a statistical sense.

Confidence-based recommender systems have been proposed based on collaborative filtering. The first system is based on group recommender systems [8], and the second one is based on matrix factorization [7]. Both systems can be directly applied for course recommendations, however, under assumptions typical for collaborative filtering. For example, plenty of data is available; the course order does not matter. In this context, we note that we propose a generic algorithm for conformal course recommendation that is not tailored to the recommendation mechanism: collaborative filtering or content-based filtering. The only requirement to apply this algorithm is to have a function that estimates the typicality of a course w.r.t. other courses taken by a student (see conformity functions in Section 5).

3. RECOMMENDATION TASK WITH CONFIDENCE

Let T be a set of topics t considered in a set C courses c . To indicate the degree of presence of topic t in course $c \in C$ we employ weight $w_{c,t}$. Topic weights $w_{c,t}$ of course $c \in C$ represents a topic model of this course. We assume that the topic models (i.e. topics' weights $w_{c,t}$) are provided initially for all the courses $c \in C$. We describe our approach to derive these models in the next Section.

The courses $c \in C$ are given for a set S of students s . To indicate the degree student $s \in S$ masters topic t we employ weight $w_{s,t}$. Topic weights $w_{s,t}$ of student $s \in S$ represents a topic model of the student w.r.t. courses $c \in C$. Thus, they are computed w.r.t. set C_s of courses student s has taken; i.e. for any topic $t \in T$ we have:

$$w_{s,t} = \frac{\sum_{c \in C_s} w_{c,t}}{|C_s|}, \quad (1)$$

If we assume a specific ordering of the topics $t \in T$, then:

- the topics' weights $w_{c,t}$ for course c form a topic-model vector w_c for c , and
- the topics' weights $w_{s,t}$ for student s form a topic-model vector w_s for s .

The topic-model vectors w_c and w_s "live" in the same space W . Due to the number $|T|$ of all the topics, we employ the

cosine similarity over W . It can be used to compute similarity for any two topic-model vectors that represent courses and students.

The topic-model vectors w_c of all the courses $c \in C$ form the course data set W_C defined as $\{w_c\}_{c \in C}$. Analogously, the topic-model vector w_t of all the students $t \in T$ form the student data set W_S defined as $\{w_t\}_{t \in T}$. In this context, we introduce the recommendation task considered in this paper. Given a course data set W_C , a student data set W_S , and a student $s \in S$ with topic-model vector $w_s \in W_S$, the task is to compute a recommendation set $C_s^\epsilon \subseteq C \setminus C_s$ that contains courses that indeed fit student s with a probability at least $1 - \epsilon$ for a predefined significance level $\epsilon \in [0, 1]$.

4. COURSE AND STUDENT TOPIC MODELING

We employed the topic-modeling approach proposed in [11]. The set T of topics t was identified from the course descriptions using the Latent Dirichlet Allocation (LDA) generative model [2]. Each topic $t \in T$ is given by a probability distribution over the vocabulary derived from all the descriptions. Thus, each course $c \in C$ is represented by topics t , which words are present in the description of that course. Student topic models are derived based on the topics courses using formula (1).

5. CONFORMAL COURSE RECOMMENDATION

This section introduces a conformal course recommendation. First, we present a conformal test for course inclusion and a generic algorithm for conformal course recommender systems. Then we provide two instantiations of this algorithm.

5.1 Generic Conformal Course Recommender

Consider a particular student $s \in S$ with her set C_s of courses. We assume that student s is represented by a probability distribution P_s ; i.e. P_s has generated the course set C_s for s . Thus, to decide whether to recommend a new course $c \notin C_s$ for student s , we perform a statistical test of the null hypothesis that the set $C_s \cup \{c\}$ is generated by the student distribution P_s under the exchangeability assumption [15]¹.

We implement the statistical test according to the conformal-prediction framework [15]. It makes use of course conformity scores. The conformity score α_c of a course c is defined as a score that indicates how typical c in set $C_s \cup \{c\}$. The conformity score α_c is computed by a course conformity function A . The latter is a mapping from $2^C \times C$ to $\mathbb{R} \cup \{+\infty\}$; i.e. it returns for any course set C_s and any course c a score α_c that indicates how typical is course c for the courses in $C_s \cup \{c\}$. Depending on the implementation of the conformity function for the course and student topic models, we can have recommender systems based on content/collaborative filtering (See the next section).

The conformity score α_c of a new course c is used as a test statistic for the null hypothesis that the set $C_s \cup \{c\}$ is generated by the student distribution P_s according to the

¹We note that the exchangeability assumption is weaker than the well-know i.i.d. assumption.

exchangeability assumption. The p -value p_c for the null hypothesis, is calculated as the fraction of the courses in $C_s \cup \{c\}$ associated with conformity scores that are equal to or smaller than α_c . The larger the value of p_c , the more likely it is to observe the value of α_c under the null hypothesis, and the more confidence we have in course c . If we set a course significance level ϵ_c (probability of the error) in range of $[0, 1]$, then the statistical test will accept the null hypothesis if $p_c > \epsilon_c$ and course c will be recommended.

If we perform the statistical test from above for student $s \in S$ over all n courses from set $C \setminus C_s$ (that s has not taken), then we can compute a recommended set of courses. We summarize this process in Algorithm 1 given below. It presents a generic conformal course recommender algorithm. Given a course significance level $\epsilon_c \in [0, 1]$ and set C_s of m courses taken by student s , the algorithm computes set $C_s^\epsilon \subseteq C \setminus C_s$ of recommended courses for student s on the chosen significance level ϵ . To decide whether to include course $c_i \in C \setminus C_s$ in set C_s^ϵ the algorithm computes the conformity score α_{c_j} for each course $c_j \in C_s \cup \{c_i\}$ using the nonconformity function A (step 4). The conformity scores α_{c_j} are used for computing the p -value p_{c_i} of course c_i (step 6). Once p_{c_i} has been obtained, course c_i is added to the set C_s^ϵ of recommended courses if $p_{c_i} > \epsilon_c$ (step 7). This process is repeated for every course $c_i \in C \setminus C_s$.

We note that the generic conformal course recommender algorithm computes valid recommendation sets C_s^ϵ for a significance level ϵ that usually is bigger than the course significance level ϵ_c (used in Algorithm 1). To explain this phenomena we follow the approach proposed in [9]. W.l.o.g. assume that the set $C \setminus C_s$ is the same for all the students $s \in S$. Let e_{c_i} be a random error variable for a course c_i from the n courses in $C \setminus C_s$. The variable e_{c_i} equals 1 if course c_i does not fit student s ; and 0, otherwise. Assume that we set the course significance level ϵ_c so that $p(e_{c_1} = 1) < \epsilon_c, p(e_{c_2} = 1) < \epsilon_c, \dots, p(e_{c_n} = 1) < \epsilon_c$. This implies that the expected number of courses incorrectly recommended, $e_{c_1} + e_{c_2} + \dots + e_{c_n}$, is bounded by $n\epsilon_c$; i.e. $\mathbf{E}[\sum_{c_i \in C \setminus C_s} e_{c_i}] \leq n\epsilon_c$. If we know number t of courses from $C \setminus C_s$ that fit student s in advance, then:

$$\frac{1}{t} \mathbf{E}[\sum_{c_i \in C \setminus C_s} e_{c_i}] \leq \frac{n}{t} \epsilon_c. \quad (2)$$

We note that $\frac{1}{t} \mathbf{E}[\sum_{c_i \in C \setminus C_s} e_{c_i}]$ is the expected error and $\frac{n}{t} \epsilon_c$ is a significance level ϵ for which validity of recommendation sets C_s^ϵ can be established. This implies:

$$\epsilon_c = \frac{t}{n} \epsilon \quad (3)$$

Thus, to guarantee valid recommendation sets $C_s^\epsilon \subseteq C \setminus C_s$ that contains courses that fit students with a probability at least $1 - \epsilon$ we need to set the course significance level ϵ_c according to formula (3) when we initialize the generic conformal course recommender algorithm from Algorithm 1.

Algorithm 1 Generic Conformal Course Recommender

Input: Course significance level ϵ_c ,
Set C_s of m courses taken by student s .
Output: Set C_s^ϵ of recommended courses for student s .

- 1: Set course set C_s^ϵ equal to \emptyset .
- 2: **for each** course $c_i \in C \setminus C_s$ **do**
- 3: **for each** course $c_j \in C_s \cup \{c_i\}$ **do**
- 4: Set conformity score α_{c_j} of course c_j equal to $A(C_s \cup \{c_i\}, c_j)$.
- 5: **end for**
- 6: Set p_{c_i} equal to $\frac{\#\{c_j \in C_s \cup \{c_i\} \mid \alpha_{c_j} \leq \alpha_{c_i}\}}{m+1}$.
- 7: Add course c_i to C_s^ϵ if $p_{c_i} > \epsilon_c$.
- 8: **end for**
- 9: Output set C_s^ϵ of recommended courses for student s .

To establish the validity of sets C_s^ϵ of recommended courses, we adapt the error metric from [9, 19]. Assume that for student $s \in S$, we have a test set of courses, and we know that within this set, there is a true set C_s^t of courses that student s will take. We define the individual error e_s for student $s \in S$ as the proportion of the courses in the true set C_s^t that are not recommended, i.e.

$$e_s = \frac{|C_s^t \setminus (C_s^\epsilon \cap C_s^t)|}{|C_s^t|}.$$

In this context, the error e of a conformal course recommender is defined as the averaged error e over all the students $s \in S$:

$$e = \frac{\sum_{s \in S} e_s}{|S|}.$$

We note that the individual error e_s corresponds to the expected error $\frac{1}{|C_s^t|} \mathbf{E}[\sum_{c_i \in C \setminus C_s} e_{c_i}]$. Thus, to show experimentally that a conformal course recommender is valid for any significance level ϵ in $[0, 1]$, we have to show that the error e is less than or equal to ϵ .

The validity of a conformal course recommender can be trivially achieved if the recommender outputs all the possible courses from set $C \setminus C_s$. Thus, we need to estimate the informational efficiency of the recommender. For this purpose we employ the size SR of the recommended set C_s^ϵ of courses averaged over all the students:

$$SR = \frac{\sum_{s \in S} |C_s^\epsilon|}{|S|}.$$

5.2 Content-based Conformal Course Recommender Systems

The generic conformal course recommender algorithm can be instantiated if we specify the course conformity function A . This function can be done using different recommender mechanisms, e.g., collaborative filtering or content-based filtering. In this paper, we assume the existence of topic model

vectors of courses and students that fit the content-based filtering scenario (see Section 3). That is why we propose conformity functions for two content-based conformal course recommender systems specified below.

The first system is an instance-based conformal course recommender system (ICCRS). Any student $s \in S$ is represented by a set of topic-model vectors (instances) w_c of the courses $c \in C_s$ she has taken. In this context the course conformity function A outputs for any course $c \in C$ and course set C_s of student $s \in S$ an averaged similarity of c with courses in C_s ; i.e. $\frac{1}{|C_s \setminus \{c\}|} \sum_{c' \in C_s \setminus \{c\}} \cos(w_c, w_{c'})$ where \cos is the cosine similarity.

The second system is an exemplar-based conformal course recommender system (ECCRS). It employs topic-model vector w_s (exemplar) of student $s \in S$ computed using formula (1). In this context the course conformity function A outputs for any course $c \in C$ and course set C_s of student $s \in S$ a value equal to $\cos(w_c, w_s)$, where topic-model vector w_s of student $s \in S$ is based on the courses in $C_s \setminus \{c\}$ and \cos is the cosine similarity.

The computational complexity of ECCRS is higher than that of ICCRS since, for any student, we need to recompute her topic-model vectors w_s by excluding courses one by one. However, ECCRS has better explanation capabilities. The topic-model vector w_s of student s represents the current levels of topic mastering, and the topic-model vector w_c , of course, c represents the topics covered in the course. Thus, the cosine match can explain why the course has been selected/rejected.

6. STUDENT-COURSE DATA

ICCRS and ECCRS have been implemented as course recommender systems for University College Maastricht. The college has provided course enrollment data from 2008 to 2017. This data includes course and student identifiers, grades for each course, details regarding course assessment, ECT credits, and course descriptions. The course descriptions facilitate the construction of topic values for both the student model and the course model. The calculation of topic values is with LDA, and an optimal number of topics is determined through maximum likelihood estimation. This optimization results in sixty-five topic areas representing the course catalog [11]. We remove modules without descriptions from consideration. In total, 143 courses and 2422 students enrolled in at least one course remain.

The rates of course enrollments vary widely between each course. Registration in the majority of courses offered occurs only a few times over the entire period, see in Figure 1. The modules provided are updated each year, reflecting the changes to the course catalog via dropping courses and course code changes. Several introductory courses, required courses, and projects make up a significant portion of all enrollments. Most students at UCM need eighteen periods to complete their education. Nevertheless, some students enroll in over twenty periods. See Figure 2. Each recommender system focuses on a subset of twelve periods representing two years at UCM. The subset is refined further by selecting only students starting in the fall intake semester. These restrictions increase the standardization of students

for our systems, and balance for the diversity of enrollment patterns present in an open-course curriculum. Our recommender systems use the remaining 1018 students that fall within these boundaries.

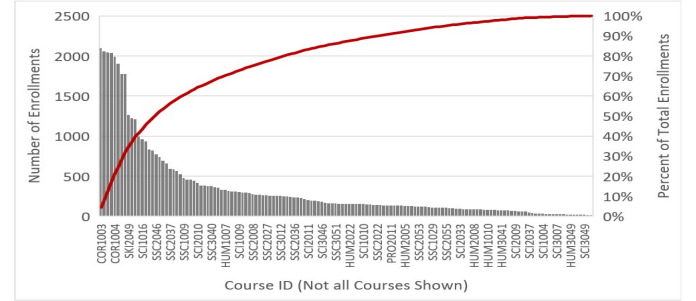


Figure 1: Total Number of Course Enrollments

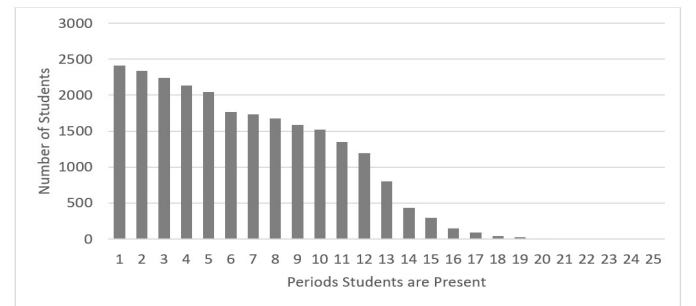


Figure 2: Total Number of Student Enrollments per Period

University College Maastricht offers a project-based curriculum. Two of the six periods each year are for student projects (periods three and six). The choice of courses within project periods is restricted. See Figure 3. Excluding these project periods, the average courses offered each period is 31 with a maximum of 44 courses. Figure 3 shows the variation in course offerings throughout the UCM data available. This variation is taken into account by our systems, and we omit these project periods from calibration. Course recommendations include only those courses offered in the target period. Therefore, recommending the maximum number of courses for a period results in a 0% error.

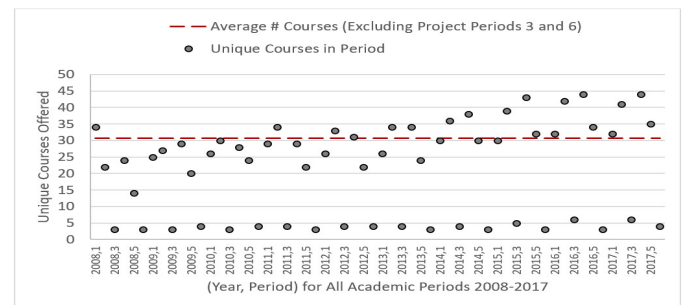


Figure 3: Courses Offered in Each Academic Period

7. EXPERIMENTS

This section presents experiments of ICCRS and ECCRS on the student-course data provided by University College Maastricht (see Section 6). First, an experimental setup is given, followed by results and discussion.

7.1 Setup

We validate ICCRS and ECCRS on the student-course data in the order of study periods. Assume that we have M number of periods P_1, \dots, P_M in which a student studies towards her degree (for our data M is equal to 18). We denote by $C_s(P_m) \subseteq C_s$ the set of courses that student s has taken in period P_m for $m < M$. Given new period P_{m+1} together with the set $\cup_{m=1}^M C_s(P_m)$ of courses student s has taken before that period, we test our recommender systems by checking whether the recommended sets C_s^ϵ of courses for P_{m+1} includes the courses $C_s(P_{m+1})$ that student s indeed has taken in P_{m+1} .

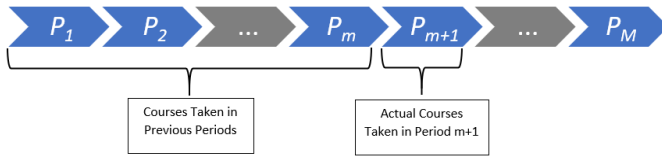


Figure 4: Prediction Process for Period P_{m+1}

We validate ICCRS and ECCRS using data from students in their second year of study. This choice is due to the fact the number of p -values possible is related to the number of courses a student has taken (see line 6 of the generic conformal course recommender in Algorithm 1). For example, a student with only three courses taken in period P_1 of year one can only have p values from the set $\{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ for any new course in period P_2 .

For the validation process, we estimate the average error of e , and the average size of SR of the recommended sets C_s^ϵ of courses. We then use these statistics to study ICCRS and ECCRS as conformal predictors and as recommender systems.

In the first study, when we investigate ICCRS and ECCRS as conformal predictors, we are interested in establishing the validity and informational efficiency of the systems (check Sub-section 5.1). In the second study, when we investigate ICCRS and ECCRS as recommender systems, we are interested in estimating the error of the systems over the periods when we employ the recommended sets C_s^ϵ on a given course significance level ϵ . In our experiments, we use course significance levels ϵ of 0.05 and 0.1.

7.2 Results and Discussion

Figures 6 and 7 present the error plots and size plots of the recommended sets C_s^ϵ of ICCRS and ECCRS, respectively, for course significance level ϵ_c ². The error curves are very close to the diagonal $(0, 0) - (1, 1)$, which means the error is close to the course significance level ϵ_c . For ICCRS, the

²We use the course significance level ϵ_c instead of the significance level ϵ for the predicted course sets since the range of ϵ is very restricted according to formula (3); e.g. $[0, 0.025]$ for 40 possible courses in a study period.

error is bounded mainly from above. For ECCRS, the error is bounded mainly from below. This bounding indicates that the systems are valid given sufficient information, especially ICCRS, which is conservatively valid [15].

The conservative validity of ICCRS explains why the averaged size SR of the recommended sets $C_s^{\epsilon_c}$ is higher than that of ECCRS. Thus, we may conclude that the informational efficiency of ECCRS is better in our experiments.

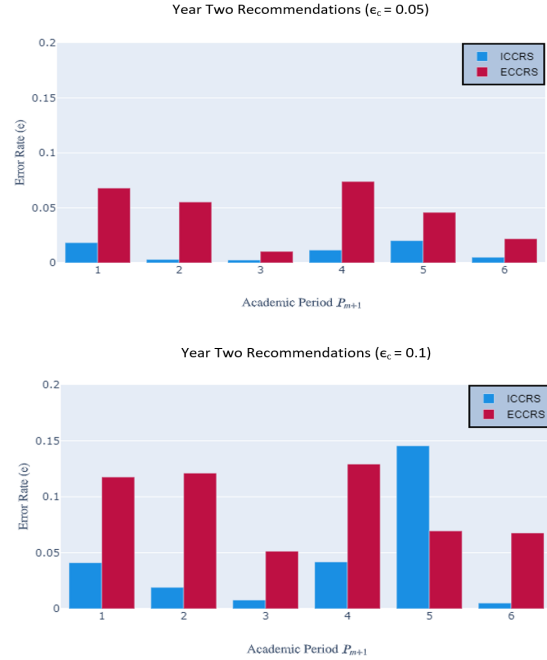


Figure 5: Period errors of ICCRS and ECCRS on course significance level ϵ_c of 0.05 and 0.1

Figure 5 presents the error of ICCRS and ECCRS when recommended sets C_s^ϵ on course significance levels ϵ_c of 0.05 and 0.1 are used. The systems are applied over periods of P_1, P_2, P_3, P_4, P_5 , and P_6 of the second year of the UCM students. The results show that:

- ICCRS and ECCRS produce accurate recommended sets of courses with an acceptable error.
- ICCRS is more accurate than ECCRS. This difference can be explained by the fact that ICCRS is more conservatively valid.
- the course significance level ϵ_c plays a substantial role: for 0.05, the error of recommended sets C_s^ϵ is much lower. However, this comes with a price: the size of the recommended sets is bigger when epsilon is lower.

8. CONCLUSION

This paper shows that safe course selection can be obtained if recommendations are supported with statistical confidence. The statistical confidence can be used for computing a statistically valid set of recommended courses that contains courses a student is likely to take with a probability of at least $1 - \epsilon$ for a user-defined significance level ϵ .

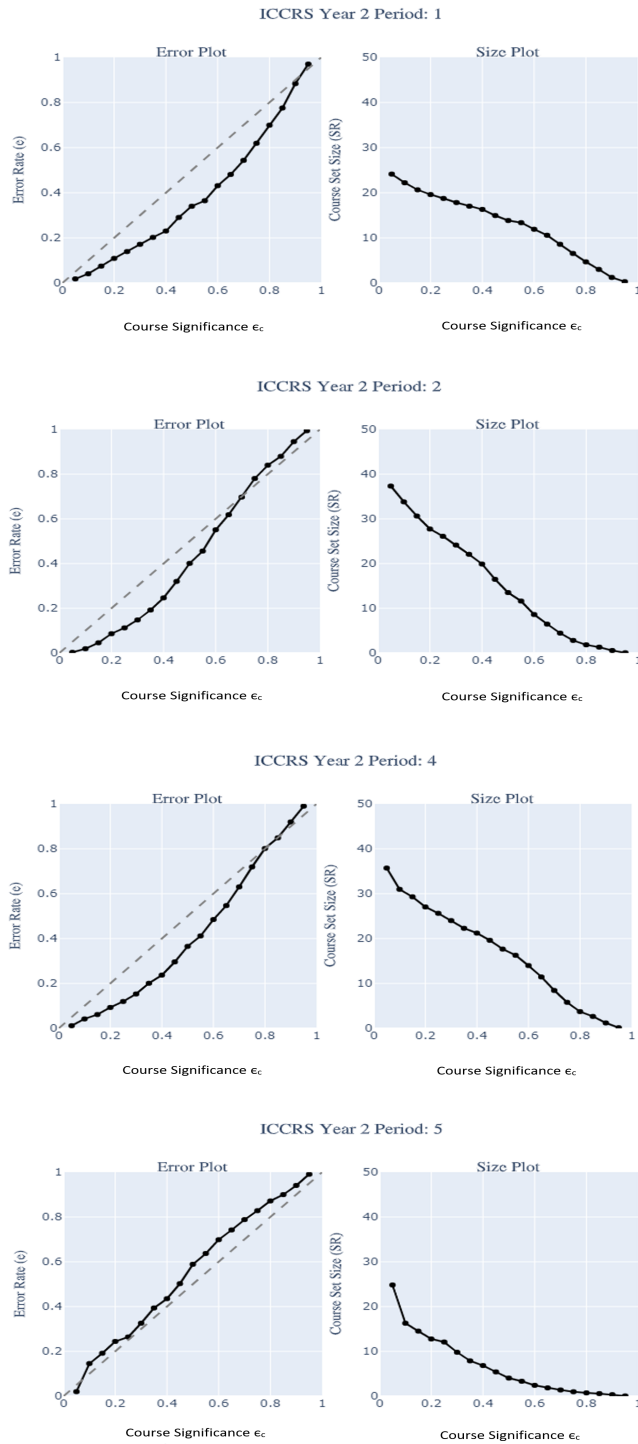


Figure 6: ICCRS - Calibration and Course Set Sizes

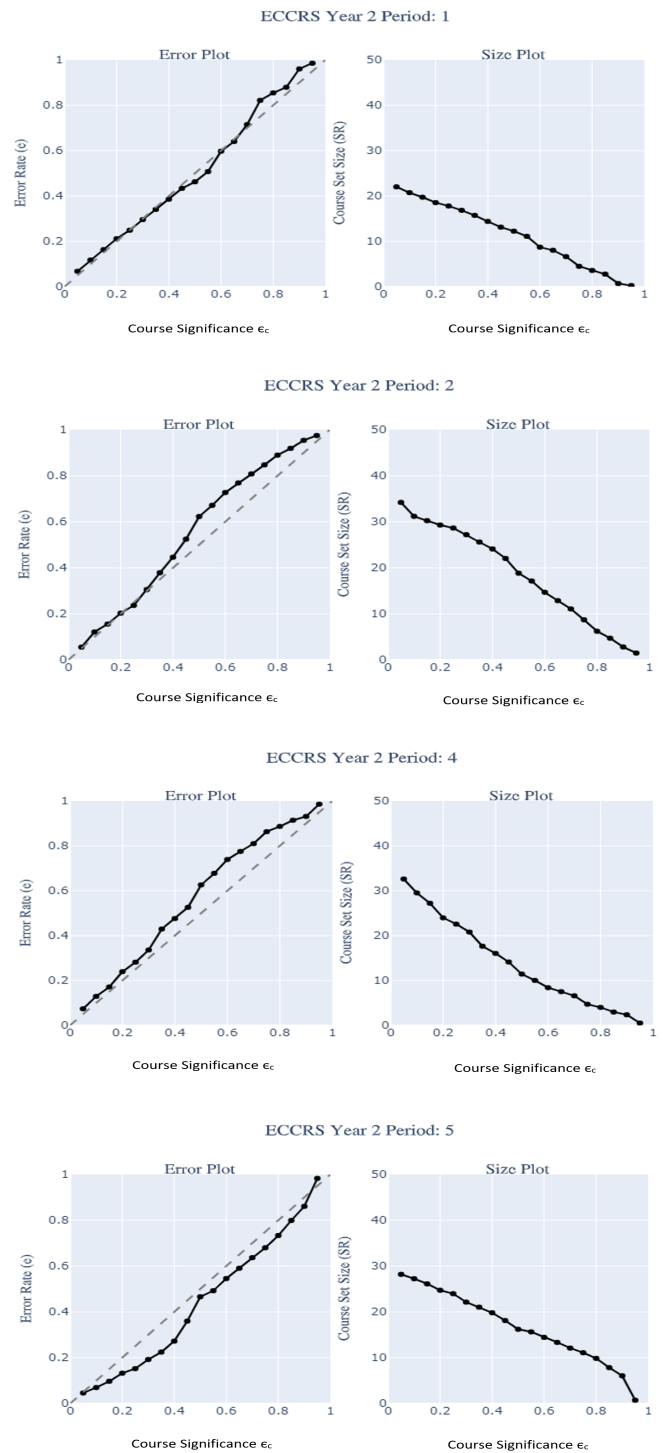


Figure 7: ECCRS - Calibration and Course Set Sizes

We have developed a generic conformal course recommender algorithm that outputs recommendations supported by statistical confidence. The algorithm has been instantiated in the form of two confidence-based course recommendation systems. The systems are essentially content-based: the first is an instance-based recommender system with rela-

tively high accuracy. The second system is an exemplar-based system with a lower accuracy but with better explanatory capabilities. The experiments showed that both systems accurately suggest courses to students while providing statistically valid sets of courses recommended.

9. REFERENCES

- [1] C. C. Aggarwal. *Recommender Systems: The Textbook*. Springer, 2016.
- [2] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- [3] H. Bydzovská. Course enrollment recommender system. In *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016*, pages 312–317. International Educational Data Mining Society (IEDMS), 2016.
- [4] H. Dai, J. Liu, and E. Smirnov (eds.). *Reliable Knowledge Discovery*. Springer, 2012.
- [5] K. Driessens, I. Koprinska, O. C. Santos, E. N. Smirnov, K. Yacef, and O. R. Zaiiane. UMAP 2017 Education Recommender Systems Workshop. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization, UMAP 2017*. ACM, 2017.
- [6] A. Elbadrawy and G. Karypis. Domain-aware grade prediction and top-n course recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, page 183–190, 2016.
- [7] T. Himabindu, V. Padmanabhan, and A. Pujari. Conformal matrix factorization based recommender system. *Information Sciences*, 467:695–707, 2018.
- [8] V. Kagita, A. K. Pujari, V. Padmanabhan, S. Sahu, and V. Kumar. Conformal recommender system. *Information Sciences*, 405:157 – 174, 2017.
- [9] A. Lambrou and H. Papadopoulos. Binary relevance multi-label conformal predictor. In *Proceedings of 5th International Symposium Conformal and Probabilistic Prediction with Applications, COPA 2016*, volume 9653 of *Lecture Notes in Computer Science*, pages 90–104. Springer, 2016.
- [10] H. Ma, X. Wang, J. Hou, and Y. Lu. Course recommendation based on semantic similarity analysis. In *Proceedings of the 3rd IEEE International Conference on Control Science and Systems Engineering, ICCSSE 2017*, pages 638–641, 2017.
- [11] R. Morsomme and S. V. Alferez. Content-based course recommender system for liberal arts education. In *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019*. International Educational Data Mining Society (IEDMS), 2019.
- [12] M. P. O’Mahony and B. Smyth. A recommender system for on-line course enrolment: an initial study. In *Proceedings of the 2007 ACM Conference on Recommender Systems, RecSys 2007*, pages 133–136. ACM, 2007.
- [13] M. Richter and R. Weber. *Case-Based Reasoning*. Springer, 2013.
- [14] A. C. Rivera, M. Tapia-León, and S. Luján-Mora. Recommendation systems in education: A systematic mapping study. In *Proceedings of the International Conference on Information Technology Systems, ICITS 2018*, pages 937–947, 2018.
- [15] G. Shafer and V. Vovk. A tutorial on conformal prediction. *J. Mach. Learn. Res.*, 9:371–421, 2008.
- [16] E. N. Smirnov and A. Kaptein. Theoretical and experimental study of a meta-typicalness approach for reliable classification. In *Workshops Proceedings of the 6th IEEE International Conference on Data Mining, ICDM 2006*, pages 739–743. IEEE Computer Society, 2006.
- [17] A. Surpatean, E. N. Smirnov, and N. Manie. Similarity functions for collaborative master recommendations. In *Proceedings of the 5th International Conference on Educational Data Mining, EDM 2012*, pages 230–231.
- [18] A. Surpatean, E. N. Smirnov, and N. Manie. Master orientation tool. In *Proceedings of 20th European Conference on Artificial Intelligence, ECAI 2012*, pages 995–996. IOS Press, 2012.
- [19] S. Zhou, E. N. Smirnov, and R. Peeters. Conformal region classification with instance-transfer boosting. *International Journal on Artificial Intelligence Tools*, 24(6):1–25, 2015.

Problem detection in peer assessments between subjects by effective transfer learning and active learning

Yunkai Xiao [yxiao28],¹ Gabriel Zingle [gzingle],¹ Qinjin Jia [qjia3],¹ Shoaib Akbar [sakbar],¹
Yang Song [songy],² Muyao Dong [1120172192],³ Li Qi [1120172633],³
and Edward Gehringer [efg]¹

¹ North Carolina State University, Raleigh, North Carolina 27695, USA [@ncsu.edu]

² University of North Carolina at Wilmington, Wilmington, NC 28407, USA [@uncw.edu]

³ Beijing Institute of Technology, Beijing 110819, China [@bit.edu.cn]

ABSTRACT

Peer assessment adds value when students provide “helpful” feedback to their peers. But, this begs the question of how we determine “helpfulness.” One important aspect is whether the review detects problems in the submitted work. To recognize problem detection, researchers have employed NLP and machine-learning text classification methods. Past studies have used datasets that were narrowly focused on a small number of classes in specific academic fields. This paper reports on how well models trained on one dataset or field perform on data from classes that are unlike the classes whose data they have been trained on. Specifically we took a model developed with data from a computer science class with several programming assignments, and tried to transfer it onto an education class focused more on writing research papers. We have attempted to perform such a task on a few models including logistic regression classifier, random forest classifier, multinomial naive bayes classifier and support vector machine. We made several attempts to raise the accuracy of classification, including lemmatizing to deduct variation in data input, and active learning strategies.

1. INTRODUCTION

The term “peer assessment” means students reviewing each other’s work. The practice has been widely used for at least fifty years. It began as a face-to-face process, with students exchanging their papers. For the last twenty-five years or so, peer assessment has also been performed using online applications. Peer assessment has many advantages. From a pedagogical point of view, the greatest advantage is that it helps students understand the requirements for the assignment, and see how their work measures up to their peers [1, 2, 3, 4, 5]. This helps them to improve their own work product. From an operational standpoint, peer assessment is scalable—no matter how many students are in the course, students’ work does not want for personal attention. This makes peer assessment especially useful for MOOCs, where

it is frequently to provide feedback and to assign grades.

Student work on a MOOC can be graded in different ways. If objective questions are posed, such as multiple-choice and true/false questions, they can be automatically graded by software that checks whether answers match the key, while for subjective issues such as coding projects and essays, it becomes a bigger challenge. These platforms often utilize quantitative methods such as averaging reviewer scores on multiple sections of peer assessment related to the course assignment.

Current peer grading approaches are based on the numerical scores assigned to rubric items by each reviewer. Rarely do they utilize another very important piece of information: the justification given by reviewers for the grades they’re giving.

Fundamentally, the quality of a review is related to whether it identifies ways for the author to improve the work. Thus, it is important for the review to point out shortcomings or problems in the existing work. Other researchers [6] have done preliminary work in this area. They have looked at approaches to detecting suggestions [7], for the reason that suggestions help students act on improving the work they have done. Other work involves recognizing problem statements. A problem statement helps people realize the shortcomings in their work, and pointing out a problem does not require as much thinking as knowing what is wrong and coming up with a solution to correct the problem as making a suggestion does. In the context of peer review, if we could tell whether a comment contains one of these features (suggestion or problem statement), we could compare a reviewer’s work with other reviewers’ and urge him/her to add more to the review if his/her review lacks these features significantly. In order to accomplish it, a means of automatically detecting these features needs to be devised.

We have built text classifiers that can recognize whether a comment contains a problem statement; however there’s a drawback. As researchers know, text classifiers are very domain specific, that is if a classifier is trained on one specific domain, it will probably not perform well when used on another domain [8]. When MOOCs offer classes in multiple fields, the peer reviews in each class will have different language features. Useful sentiment features such as problem statements would not be the same in different classes. Traditionally, there would be multiple classifiers trained on each

Yunkai Xiao, Gabriel Zingle, Qinjin Jia, Shoaib Akbar, Song Yang, Muyao Dong, Li Qi and Edward Gehringer “Problem detection in peer assessments between subjects by effective transfer learning and active learning” In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 516 - 523

one of the domains to achieve optimal performance. An issue with this approach is that there needs to be enough labeled data from each of these domains, which in a lot of cases is hard to achieve. Labeling is becoming one of the most expensive steps in machine learning, both from the perspective of time and of money [9]. However, there are a number of ways to work around this problem, if not completely mitigate it.

Researchers have demonstrated that traditional machine learning and deep learning technologies are useful for problem detection in peer review in the computer science field [10]. The researchers aim to generalize the problem detection function to different subjects. There are two potential methods for quickly building a model in a target domain and avoiding much of the time-consuming and expensive data labeling efforts. Such methods include transfer learning and active learning. With these two approaches, problem detection could be transferred quickly to a new field and at a reduced cost.

The first approach is to leverage transfer learning to transfer “knowledge” learned from the problem detection task in the computer science field to the other field. This process can use model insights gained from other datasets to expedite the construction of a new model while including only a small amount of labeled data in the target domain. In our case, we trained a problem detection classifier from data generated in a computer science class. One of the research questions we aim to discuss is leveraging transfer learning to effectively preserve the performance of the model when it is applied to other classes.

The second method is to utilize active learning to label abundant data and then apply machine learning algorithms or train deep neural network models on this automatically labeled data. This method is detailed in the implementations subsection of the experiment section of this paper.

2. LITERATURE REVIEW

2.1 Problem Detection

There have been plenty of attempts to apply natural language processing (NLP) techniques and machine-learning (ML) algorithms on automating various aspects of review assessment. Brun and Hagege [11] leveraged NLP techniques to identify suggestions in review text. Zingle et al [7] attempted to use different ML and Deep learning algorithms to determine whether a review text contains suggestions. Nguyen et al. [12] used logistic regression to train a model that predicted whether a review comment contained a problem solution. They provided this information to the reviewer before the review was submitted, in order to encourage the reviewer to suggest solutions for problems in the work.

However, most of the current research related to applying NLP and ML on peer review is limited to one subject or ones filled with enough labeled data. For example, research from Zingle et al. [7] collected student annotated peer reviews from a graduate level computer science course and used this labeled data to train models for detecting suggestions in the course. The study by the Brun and Hagege [11] did similarly with abundant manually annotated customer reviews. To the best of our knowledge, there are no pub-

lished papers that address the issue of how to apply NLP and ML on peer reviews in a field without abundant labeled data. This paper is based on previous research about detecting problem statements in peer assessments [10]. This paper focuses on detecting problem statements in a field without abundant labeled reviews by utilizing transfer learning and active learning.

2.2 Transfer Learning

In most traditional machine learning algorithms, an essential hypothesis is that the training data and test data must be in the same feature space and have the same distribution [13, 14]. If the feature space or latent distribution changes, sufficient labeled data from the new domain will be needed and the statistical model must be rebuilt from scratch. This approach can be time-consuming and expensive in many real-world applications like text classification and thus limits its development [15]. The peer-review comments from the computer science field and the peer-review comments from other subjects might be in the same feature space but in different distribution, where plenty of peer-review comments from each field must be labeled and a learner must be reconstructed from scratch for each subject.

In contrast, transfer Learning, which is fundamentally motivated from a discussion in a NIPS-95 workshop [16], relaxes the hypothesis that the training data must be in the same feature space and identically distributed with the test data [13, 14]. The basic idea of transfer learning is to transfer “knowledge” learned from source tasks to different but related target tasks. This is to combat against the problem of an insufficiently large labeled training dataset and to improve the learning of the target task by reducing the labeling cost. In this case, only a small quantity of labeled data in the target domain is required. Negative transfer may occur, but a successful “transfer” would greatly improve the performance and reduce the cost of learning for the target task by avoiding much time-consuming and expensive data labeling efforts.

Pan and Yang [13] summarized various transfer learning settings and categorized transfer learning under three sub-settings. These include inductive transfer learning, transductive transfer learning, and unsupervised transfer learning, based on different situations between the source and target domains and task. This paper is under the inductive transfer learning setting, which has different, yet related source and target domain tasks, where a sufficient quantity of labeled data is only required in the source domain. There are five main approaches for conducting the inductive transfer learning from literature. These approaches are instance-based transfer learning [17, 18], feature-representation transfer [19, 20], parameter-transfer [21, 22], relational-knowledge-transfer problem [23, 24], and Hybrid-based (instance and parameter) transfer learning [25, 26].

The parameter-transfer approach mentioned above is a simple but effective method for transferring “knowledge” by sharing parameters. Assumption of the approach is that some parameters are shared by source tasks and target tasks [13]. The “knowledge” is encoded into and transferred across tasks by those shared parameters.

2.3 Active Learning

Active Learning is a significant subfield of machine learning and a helpful technique in many real-world applications where there is abundant unlabeled data, but where labels are difficult, time-consuming, or expensive to acquire [27]. Active learning algorithms are allowed to interactively query a human annotator called teacher or oracle to label the new data point chosen by a predefined strategy and usually perform better with less labeled training data. There are three main settings in which the learner may be able to query. These settings are membership query synthesis proposed by Angluin [28], stream-based selective sampling proposed by Cohn et al. [29] and pool-based active learning proposed by Lewis and Gale [30].

The most common active learning scenario is the pool-based active learning setting, which assumes that there is a smaller set of labeled data and a large pool of unlabeled instances. The key hypothesis of pool-based active learning is that the learning algorithm would perform better with less training if the algorithm could determine which instances in the pool are the most informative and is allowed to ask queries based on a certain query strategy. This would be in the form of unlabeled instances that are to be labeled by an oracle (e.g. a human annotator) [27, 30]. Hoi and et al. [30] investigated the pool-based scenario on large-scale text classification and first demonstrated the feasibility of batch mode pool-based setting active learning on the text categorization problem.

Under each active learning scenario, there have been a number of query strategies proposed for evaluating the informativeness of unlabeled instances. We evaluated the most common query strategies, uncertainty sampling published by Lewis and Gale [30]. The uncertainty sampling strategy selects the instance in the pool about which model is least certain on how to label observations according to an uncertainty measure like entropy.

In contrast to active learning, traditional passive learning would use a random sampling strategy to select instances from a large pool of unlabeled instances. This strategy generally underperforms compared with uncertainty sampling strategy thus is not adopted here.

3. EXPERIMENT

3.1 Data

To train the problem statement classifier, we used a dataset pulled from the Expertiza system. Expertiza is a web based education platform instructors can use to distribute homework assignments and team projects. The key feature of this platform comes in later stages once students submit their work, where they assess the work product of other students by giving a numeric score as well as a comment to justify their decision. For team assignments, students would assess work done by other teams, as well as the contributions of their teammates.

In some of the classes, students are asked to annotate the comments they received with an incentive of extra credit with a “yes” or “no” on given metrics. For example, some metrics that the students label for include “Does the comment contain a problem statement?”, “Does the comment offers a suggestion?”, or “Was the comment localized to a

particular place in the work?”. This is a valuable source of annotated data for our research, as students should be experts at annotating feedback on their own work. However, many times more steps are required to improve the quality of this data. On observing the annotations, we found a number of problems. Sometimes students would rush through the annotation with the goal of getting extra credit with minimal effort, leaving a trail of yes’s or no’s without actually reading the comments. Other times fatigue may set in while annotating a large number of comments, resulting in the accuracy of labels gradually dropping towards the end of the annotation process. To resolve this issue, the course staff and the research team checks labels applied by the students through random sampling of the students’ annotations. If it appeared that a student was not taking adequate care, that student’s annotations would be removed from the dataset.

We extracted data from computer science class projects. Since every member of the team is involved in annotating reviews they received for team projects, we were able to calculate inter rater reliability using Krippendorff’s alpha, which was relatively low at a value of 0.696. To improve the accuracy of our model, we decided to only take those data with consensus among all annotators, by removing those with any conflicted labels, which decreased the size of our dataset by 4649, resulting in an improved Krippendorff’s alpha of 1. We then further altered the dataset by downsampling the majority class by 313 observations to ensure a balanced proportion of classes.

To prove that language features, specifically for problem statements in this particular dataset could be transferred, we run a test on three other datasets. The first composes Hotel product reviews, the second Amazon reviews, and the third a small dataset from a university level education class. The Hotel and Amazon datasets were found on the website Kaggle, which states that the data originated from the website Datafiniti. Two useful columns from the original datasets included a review score from the original 1 to 5 scale, with 1 being very bad to 5 being very good, and a column with the actual review text. From inspecting the data, we found that reviews with low ratings mentioned problems regarding the respective hotels or amazon products they were reviewing, while there was no mention of a problem in well rated reviews. Based on this information, we kept all the reviews with a rating of 1 or 2 and relabeled them all to the value 1 to represent that these reviews mentioned a problem. We then kept an equal quantity of positive reviews, all labeled 5, and relabeled these to the value 0 to represent that these reviews did not mention a problem.

The target domain dataset that we’re primarily trying to transfer is generated from the education class, which had been taught using the Expertiza system. The nature of assignments in this particular class involves much more writing in terms of research papers as compared with the project based assignment in the computer science class. Students in this class are not asked to annotate the feedback reviews they’ve received, thus creating an issue in terms of a lack of labeled data. Different members of the research team did some manual inspection and annotation on small subsets of this data, then removed those data entries with conflicting labels to reach a complete consensus. This dataset was man-

ually labeled by our research team as either 1 mentioning a problem, or 0 not mentioning a problem.

We started by preprocessing the text in all four of the datasets. Specifically, we removed all punctuation aside from sentence ending period marks. We then removed all special characters and numbers. We removed URL links and converted the text to lowercase. Afterwards, we decided to balance the datasets using downsampling in terms of class proportion for observations mentioning and not mentioning a problem. This helps with models, particularly Naive Bayes, to prevent overpredicting a class based on the proportion of training data of a certain class instead of the input features. However, we did not balance the Education dataset since it was not being used to train the models and due to its small size. The total number of observations in the Expertiza, Hotel, Amazon, and Education datasets were 18354, 4460, 2442, and 172 (122 labeled 0 and 50 labeled 1) respectively.

Additionally, we have attempted to apply lemmatization and stopword removal to gauge its impact on model performance. The intuition of this is with lemmatization, we would reduce the variation of data embedding, helping the models to focus on important features to achieve better results.

3.2 Models

Before we could transfer knowledge into models that work in the target domain, some machine learning from the source domain is required. For this task, we pick four models including the Random Forest classifier, multinomial naïve Bayes classifier, support vector machine, and logistic regression classifier. Each classifier used the same 90-10% train-test split with hyper parameters tuned using 5-fold cross-validation.

Leveraging the power of the Scikit-learn package, we were able to build a data pipeline for this task [31]. Cleaned data was funneled into a count vectorizer, then weight transformed with a TF-IDF transformer, before being used by the classifiers.

The logistic regression classifier uses a regression equation to produce discrete binary outputs through a sigmoid function. It learns the coefficients of each input feature through the fitting process just like in linear regression.

The random forest classifier uses an ensemble approach that fits multiple decision trees, then uses averaging to improve the accuracy of predictions as well as to avoid overfitting. The loss criterion to choose from includes gini and entropy.

The multinomial naïve Bayes classifier is a special instance of a naïve Bayes classifier that follows a multinomial distribution for each feature $p(f_i|c)$. The naïve Bayes model assumes that each of the features it uses for classification are independent of one another.

The support vector machine classifier works by establishing a decision boundary as well as a positive plane and a negative plane between classes. Anything in the positive plane is considered to have the characteristic under study. In our experiment, this is the presence of a problem in a reviewer's comment.

We have also attempted doing the same task with a neural-network based model. One popular network structure in natural language processing is the Long Short Term Memory (LSTM) network. The LSTM takes the cleaned dataset as input, then using GloVe [32] embedding as a feature extractor before feeding them into a stacked LSTM and dense layers.

LSTM is a variation of Recurrent Neural Network (RNN), with the modification of adding the functionality of forgetting information when new information is fed into the network. This particular network leverages existing advantages of memorizing information through timesteps, and in the meantime uses four gates to input, forget, update, and output information.

3.3 Implementations

To validate our ideas on if detecting problem statements could be transferred, we did some initial experiments by training models on one dataset and then test on another. Results of these experiments could be found in the following section of the paper, where we did observe signs of knowledge being transferred and proceeded to the next stage on improving model accuracy on new domains.

Apart from transferring existing knowledge from other domains, the other way to diminish the impact of lacking annotated data is active learning. Active learning helps researchers to lessen the effort annotation by selecting a subset of high value data to annotate. Different active learning strategies may generate different subsets of data, but the essence of doing so is that it would pick data that can bring more knowledge to the models compared with other data points.

During the active learning phase, we attempted applying uncertainty sampling strategy to actively learn the more important groups of data-points listed by each model respectively. Unlabeled data from the education class dataset is exposed to all four models, and they would go through predicting whether a problem statement is present in a comment, generating labels of 1's and 0's as well as corresponding confidence scores. Using the score, we could retain four subsets of data points of which the models' confident scores are between 49% and 51%.

Two researchers then annotate over 100 of these data-points per subset, then remove conflicted entries, leaving 100 labeled data-points which each of these models are "curious" about. These observations are then appended to the computer science dataset which we originally trained the models with. Finally, the four models were re-trained separately.

4. RESULTS

In Tables 1, 2, 3, and 4 the rows represent the dataset that was used to train the model. The columns represent the dataset that was tested on by the model. In the cases marked by the diagonal in the tables, we trained the models using 90% of the dataset and tested on the remaining 10%. The order of the sets of three values within each represent the results without any further text preprocessing, lemmatization, and stopword removal respectively.

Table 1: F1 Score Logistic Regression

TrainTest	Computer Science	Hotel	Amazon	Education
Computer Science	0.89 / 0.89 / 0.83	0.70 / 0.69 / 0.68	0.70 / 0.71 / 0.63	0.73 / 0.69 / 0.64
Hotel	0.68 / 0.68 / 0.55	0.94 / 0.93 / 0.94	0.82 / 0.85 / 0.8	0.65 / 0.63 / 0.59
Amazon	0.60 / 0.58 / 0.47	0.78 / 0.81 / 0.76	0.95 / 0.93 / 0.93	0.65 / 0.63 / 0.53

*without preprocessing / with lemmatization / with stopword removed

Table 2: F1 Score Random Forest

TrainTest	Computer Science	Hotel	Amazon	Education
Computer Science	0.88 / 0.89 / 0.82	0.62 / 0.60 / 0.62	0.66 / 0.65 / 0.57	0.68 / 0.65 / 0.66
Hotel	0.74 / 0.74 / 0.59	0.91 / 0.90 / 0.92	0.73 / 0.74 / 0.73	0.61 / 0.63 / 0.60
Amazon	0.58 / 0.54 / 0.43	0.73 / 0.75 / 0.72	0.91 / 0.93 / 0.91	0.62 / 0.55 / 0.50

When models are trained on one dataset and tested on another dataset without any prior knowledge for the target domain, we could expect some drop in performance. As we tested each model's performance on different datasets, we validated this claim and found that the degradation of model performance is closely related to how much domains differ from each other.

For example, when we initially tested if something constituted a problem statement that was learned from the computer science could be transferred to other domains, we found that despite a drop of 0.2 - 0.3 in F1 score, each model did receive a F1 score larger than 0.6 for most of the runs, which is better than the random guessing average of 50%. This is a sign of positive transferring of knowledge, thus proving our idea could work.

Apart from the naive Bayes classifier, we received good results when testing on the Education dataset. This could be caused by the nature of reviews towards computer science sharing more similarities with the education dataset since they are both done by students towards their peers, unlike the other two. Apart from that, we found the knowledge transferring to the Amazon dataset constantly outperforming knowledge transferring to the Hotel dataset. When closely observing the content of the Amazon dataset, we found it is focused on reviewing electronic devices such as Amazon Kindle and Kindle fire. The nature of such projects do share some similarity with reviewing an application built by a computer science student, and as expected we could find knowledge transferred better from a computer science class to Amazon reviews compared with those from the Hotel dataset. All of the findings above can be found in Tables 1, 2, 3, and 4. Unsurprisingly, when we compare transferring knowledge between different domains through datasets that we have acquired, it can also be found that transferring works the best between the two commercial review datasets, being Amazon and Hotel, due to their nature being customer rather than peer reviews.

We analyzed the most important features most models used for prediction by examining feature coefficients from these models. The results of this examination also aligned with our observations. Within the top 20 positive and negative coefficients, we found 5 pairs of shared features between the computer science dataset and Amazon dataset. We also found 6 pairs between the computer science dataset and

Hotel dataset. Furthermore, there were 7 pairs of shared features between the Amazon dataset and Hotel dataset.

The models resulted in similar performances with and without the use of lemmatization for training and testing on the same dataset. Lemmatization did increase the accuracy when models were trained on the Hotel dataset and tested on the Amazon dataset, and vice versa. However, stopword removal led to a significant decrease in classifier performance in all cases except for when the models were trained and tested on the same dataset for the Hotel and Amazon dataset, in which case the performance was around the same.

The logistic regression classifier and support vector machine led to the best results when training and testing on the same dataset, with the exception of multinomial naive bayes when using the Hotel dataset. Otherwise, the multinomial naive bayes classifier performed the worst, particularly when attempting to predict observations found in the Education dataset.

When tested and trained on the same dataset, the models performed well with f1-scores ranging from mid 80s to mid 90s.

To bring up the accuracy when we transfer a model onto another domain, we did some active learning attempts. By using the uncertainty sampling strategy, each of the four models were exposed to the unlabeled education dataset, then the top hundred data points denoted unsure by each model is extracted. Each of these data points had a confidence between 49% and 51%, and were presented to an oracle (human annotator) for labeling. After removing conflicting labels, these subsets of data were appended to the original computer science dataset individually based on which model mentioned the uncertainty, then used to retrain each model respectively.

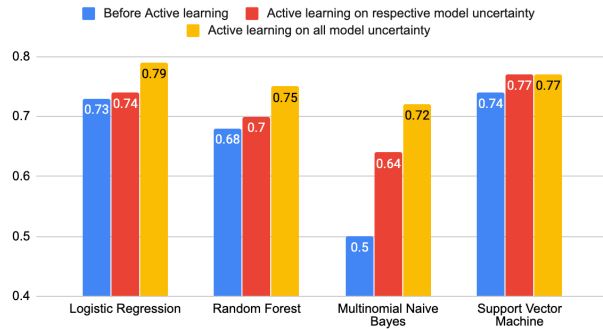
We found that with a very small carefully picked set of data, we could regain a considerable amount of accuracy after transferring a model onto a new domain. As could be seen in Figure 1, which details the affects of adding the target domain data from active learning to the computer science dataset, all models gained accuracy with Naive Bayes benefiting the most from this process.

Table 3: F1 Score Naive Bayes

TrainTest	Computer Science	Hotel	Amazon	Education
Computer Science	0.86 / 0.85 / 0.80	0.55 / 0.55 / 0.53	0.56 / 0.58 / 0.54	0.50 / 0.55 / 0.53
Hotel	0.56 / 0.53 / 0.50	0.95 / 0.95 / 0.93	0.79 / 0.82 / 0.78	0.57 / 0.54 / 0.53
Amazon	0.59 / 0.55 / 0.52	0.80 / 0.82 / 0.77	0.94 / 0.93 / 0.94	0.57 / 0.57 / 0.55

Table 4: F1 Score Support Vector Machine

TrainTest	Computer Science	Hotel	Amazon	Education
Computer Science	0.90 / 0.90 / 0.83	0.69 / 0.68 / 0.67	0.69 / 0.70 / 0.63	0.74 / 0.69 / 0.64
Hotel	0.66 / 0.66 / 0.56	0.93 / 0.94 / 0.94	0.83 / 0.85 / 0.80	0.65 / 0.62 / 0.61
Amazon	0.63 / 0.59 / 0.48	0.79 / 0.80 / 0.75	0.94 / 0.93 / 0.93	0.67 / 0.66 / 0.53

Active learning results**Figure 1: F1 Improvements with Active Learning**

There are also a few things we noticed that did not work. Ordinary data preprocessing techniques such as lemmatizing and mainly stopword removal actually reduced model performance in terms of accuracy on all four models. From reviewing the coefficients, we found that many times the tense and plurality of words actually matters, let alone a lot of the stop words. For example auxiliary verbs such as “could” and “should” often implies a problem needs to be corrected, and words implying contrast such as “but” and “however” are used to bring up readers’ attention before mentioning dissatisfaction. When these elements of language are removed, predicting whether a comment contains a problem becomes harder.

Apart from this, attempts on generating uncertain data from Neural network models and then re-train itself with resolved uncertainty does not show significant differences compared with training itself on more randomly selected samples. Results for both approaches have a F1 score fluctuate between 0.69 and 0.71 without significant differences. This could be because each time a neural network is trained, it restructures itself in a different way. With each perceptron (neuron) being a small classifier by itself, what is used to carry important knowledge to one network state might not hold as much value when the network is in a new state.

5. CONCLUSIONS AND FUTURE WORK

In conclusion, we could use models trained on one domain that classify certain sentiment components on other domains. We have tested doing problem detection between two dis-

tinctively different classes, and are confident about detecting other useful things such as suggestions or problem localizers. Results in the previous section have presented that with very little human intervention, each of the classifiers could regain a significant amount of its accuracy.

This is a very important step if we are to build a system that could promote students writing better reviews in different domains and different class settings. Furthermore, if we are to automate the grading process by involving inputs from peer assessment, we would certainly want to use features such as “how many suggestions are made” or “how many problems did the reviewer find” to gauge the quality of peer grading. Being able to analyze these features across peer assessments from different subjects becomes increasingly important.

Within this article, we mainly focused on transfer learning on traditional machine learning techniques, while there are many deep transfer learning techniques which could be utilized. With smaller datasets they might not have made much difference in terms of model accuracy. However, other researchers have shown that using layers in these neural networks trained on one dataset could be used as feature extractors for another. Examples of this are GloVe [32] and BERT [33], where both of these models are trained on a much larger dataset, resulting in exposure to a variety of knowledge, then later repurposed as feature extractors for other tasks.

In the future, we plan to explore the possibility of using transfer learning and active learning on neural network models and to continue building a review helpfulness evaluator across different subjects. In the long run, we would like to create a system that automatically assigns grades to students based on both numerical and textual peer assessments.

6. REFERENCES

- [1] Hongli Li, Yao Xiong, Charles Vincent Hunter, Xiuyan Guo, and Rurik Tywoniw. Does peer assessment promote student learning? a meta-analysis. *Assessment & Evaluation in Higher Education*, pages 1–19, 2019.
- [2] Kristi Lundstrom and Wendy Baker. To give is better than to receive: The benefits of peer review to the reviewer’s own writing. *Journal of Second Language Writing*, 18(1):30–43, 2009.
- [3] Yasemin Demiraslan Çevik. Assessor or assessee?

- investigating the differential effects of online peer assessment roles in the development of students' problem-solving skills. *Computers in Human Behavior*, 52:250–258, 2015.
- [4] Lan Li, Xiongyi Liu, and Allen L Steckelberg. Assessor or assessee: How student learning improves by giving and receiving peer feedback. *British Journal of Educational Technology*, 41(3):525–536, 2010.
 - [5] Esther Van Popta, Marijke Kral, Gino Camp, Rob L Martens, and P Robert-Jan Simons. Exploring the value of peer feedback in online learning for the provider. *Educational Research Review*, 20:24–34, 2017.
 - [6] Kwangsu Cho. Machine classification of peer comments in physics. In *Educational Data Mining*, 2008.
 - [7] Gabriel Zingle, Balaji Radhakrishnan, Yunkai Xiao, Edward Gehringer, Zhongcan Xiao, Ferry Pramudianto, Gauraang Khurana, and Ayush Arnav. Detecting suggestions in peer assessments. In *Proceedings of the 12th International Conference on Educational Data Mining*, pages 474–479. International Educational Data-Mining Society, 2019.
 - [8] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.
 - [9] P. Perona P. Welinder. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *CVPR*, 2010.
 - [10] Xiao Yunkai, Gabriel Zingle, Qinjin Jia, Harsh Shah, Yi Zhang, Tianyi Li, Mohsin Karovaliyya, Weixiang Zhao, Yang Song, Jie Ji, Ashwin Balasubramaniam, Harshit Patel, Priyanka Bhalasubramanian, Vikram Patel, and Edward Gehringer. Detecting problem statements in peer assessments. In *Proceedings of the 13th International Conference on Educational Data Mining*. International Educational Data-Mining Society, 2020.
 - [11] Caroline Brun and Caroline Hagege. Suggestion mining: Detecting suggestions for improvement in users' comments. *Research in Computing Science*, 70(79.7179):5379–62, 2013.
 - [12] Huy Nguyen, Wenting Xiong, and Diane Litman. Instant feedback for increasing the presence of solutions in peer reviews. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 6–10, 2016.
 - [13] SJ Pan and Q Yang. A survey on transfer learning. *IEEE transactions on knowledge and data engineering*, 2010.
 - [14] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*, pages 270–279. Springer, 2018.
 - [15] Hal Daume III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126, 2006.
 - [16] Jonathan Baxter, Rich Caruana, Tom Mitchell, Lorien Y Pratt, Daniel L Silver, and Sebastian Thrun. Learning to learn: Knowledge consolidation and transfer in inductive systems. In *NIPS Workshop*, http://plato.acadiau.ca/courses/comp/dsilver/NIPS95_LTL/transfer.workshop, 1995.
 - [17] J Quiñero Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. Dataset shift in machine learning. *The MIT Press*, 1:5, 2009.
 - [18] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, 2007.
 - [19] Chuen-Kai Shie, Chung-Hisang Chuang, Chun-Nan Chou, Meng-Hsi Wu, and Edward Y Chang. Transfer representation learning for medical image analysis. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 711–714. IEEE, 2015.
 - [20] Yi Zhu, Xuegang Hu, Yuhong Zhang, and Peipei Li. Transfer learning with stacked reconstruction independent component analysis. *Knowledge-Based Systems*, 152:100–106, 2018.
 - [21] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Residual parameter transfer for deep domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4339–4348, 2018.
 - [22] Chao Chen, Boyuan Jiang, and Xinyu Jin. Parameter transfer extreme learning machine based on projective model. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.
 - [23] Donghui Wang, Yanan Li, Yuetan Lin, and Yueting Zhuang. Relational knowledge transfer for zero-shot learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
 - [24] Ruidong Wu, Yuan Yao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 219–228, 2019.
 - [25] Rui Xia, Chengqing Zong, Xuelei Hu, and Erik Cambria. Feature ensemble plus sample selection: domain adaptation for sentiment classification. *IEEE Intelligent Systems*, 28(3):10–18, 2013.
 - [26] Rui Xia and Chengqing Zong. A pos-based ensemble model for cross-domain sentiment classification. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 614–622, 2011.
 - [27] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.
 - [28] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.
 - [29] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.
 - [30] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings 1994*, pages 148–156. Elsevier, 1994.

- [31] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.
- [32] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

DYNAMIC KNOWLEDGE EMBEDDING AND TRACING

Liangbei Xu
Georgia Institute of Technology
lxu66@gatech.edu

Mark A. Davenport
Georgia Institute of Technology
mdav@gatech.edu

ABSTRACT

The goal of knowledge tracing is to track the state of a student's knowledge as it evolves over time. This plays a fundamental role in understanding the learning process and is a key task in the development of an intelligent tutoring system. In this paper we propose a novel approach to knowledge tracing that combines techniques from matrix factorization with recent progress in recurrent neural networks (RNNs) to effectively track the state of a student's knowledge. The proposed *DynEmb* framework enables the tracking of student knowledge even without the concept/skill tag information that other knowledge tracing models require while simultaneously achieving superior performance. We provide experimental evaluations demonstrating that DynEmb achieves improved performance compared to baselines and illustrating the robustness and effectiveness of the proposed framework. We also evaluate our approach using several real-world datasets showing that the proposed model outperforms the previous state-of-the-art. These results suggest that combining embedding models with sequential models such as RNNs is a promising new direction for knowledge tracing.

Keywords

Knowledge tracing, Recurrent neural networks, Matrix factorization, Matrix completion

1. INTRODUCTION

A central component in many computer-based learning systems, and in any kind of *intelligent tutoring system* (ITS), is a method for estimating and tracking a student's knowledge or proficiency based on the student's previous interactions with the system. For example, a student may interact with many different course materials (homework exercises, quiz/exam questions, textbooks and other course materials, etc.) over a potentially long period of time. As a result of these interactions (as well as other external factors) the student's knowledge and proficiency will dynamically evolve

over time [3, 13, 1, 12]. Tracking the state of a student's knowledge as it evolves can provide deeper understanding how the student is learning and which interactions (questions, textbooks, etc.) are most helpful, ultimately enabling the creation of a personalized learning environment tailored to provide an improved learning experience for the student.

Estimating student knowledge or proficiency from a sequence of student interactions poses two fundamental challenges. First, student proficiency evolves over time as the student interacts with the system. For example, the student might turn to textbooks in response to getting a particular question wrong, and then may be able to answer a similar question correctly afterwards. Alternatively, the student may gradually lose proficiency in some areas if long periods of time pass without using this knowledge (e.g., over long vacations). Thus, we cannot treat this as a static problem of estimating a student's knowledge, but must think of this as a dynamic tracking problem. A second and more subtle challenge is posed by the fact that the manner in which student proficiency evolves may be strongly influenced by the nature of the interactions. For example, when a student is posed a question that requires knowledge of a particular concept, we not only learn something regarding the student's proficiency, but the student may also learn something from the question. In this way, the interactions both provide information to help us track the student's knowledge while simultaneously inducing changes in the state that we wish to track.

In this paper we propose a framework for tracing student knowledge using only a sequence of student responses to questions (for an ensemble of many students). The framework consists of two core components: a (static) embedding network that learns fixed latent representations of questions from student-question interactions and a recurrent neural network (RNN) that dynamically tracks the hidden state corresponding to each student's knowledge over time from the student's sequence of interactions. Our main contributions are:

- A new knowledge tracing framework which exploits both the advantages of latent question embedding from response data and an RNN to track student knowledge;
- A framework that can track student knowledge without using the question-level concept/skill tags that other knowledge tracing models (e.g., DKT [13] and its variants) require, avoiding labor-intensive manual tagging;

Liangbei Xu and Mark Davenport "DYNAMIC KNOWLEDGE EMBEDDING AND TRACING" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 524 - 530

- A flexible framework that can also accommodate a variety of sequential modeling techniques (e.g., memory networks [28]) and can incorporate tag information and other features when available.

2. RELATED WORK

2.1 Educational data mining

Extracting useful information from the kind of educational data we consider was first studied within the *intelligent tutoring* community. Since the seminal work of [3], there has been a variety of efforts aimed towards understanding the cognitive processes that are most relevant in the context of an ITS, most of which aim to estimate students' proficiency based on their past interactions with the system with the aim of predict their performance on the new exercises/tests or customizing their learning materials.

Static models. Item Response Theory (IRT) is a standard framework for modeling student responses to questions dating back to the 1950s [20]. Perhaps the most common IRT model is the Rasch model [15]. This is a simple two-parameter model in which each student is modelled as having a particular skill level and each question has a particular difficulty, which is then paired with a logistic link function to provide predictions of the probability a student will answer a question correctly. There are natural multidimensional extensions of this and similar IRT models, which can be viewed as special cases of standard matrix factorization models ([19]) or more general factorization machine models [16]).

Sequential models. Most of the models described above involve estimating a fixed student-question embedding which is then used to predict future responses. However, we fully expect the state of a student's knowledge to change over time. To capture such dynamics, a natural approach is to incorporate dynamics in the model. One of the most popular models is Bayesian Knowledge Tracing (BKT), which employs a hidden Markov model ([3]) to model the process of mastering a particular skill. However, the BKT approach has some significant drawbacks. Most significantly, it models only a single skill or concept at a time. In practice, any particular question may be associated with a complex combination of different skills. To overcome this shortcoming, several alternative approaches have recently been proposed.

The most relevant attempt in this direction is the Deep Knowledge Tracing (DKT) framework [13]. The DKT approach was inspired by recent progress in RNNs and deep RNN architectures. RNNs are a family of neural networks tailored for sequential prediction problems [22]. In recent years deep RNN architectures have been shown to outperform many classical models in many application areas, including natural language processing and session-based recommendation system. DKT is the first model to use RNNs to track student knowledge. DKT uses a one-hot encoding of skill/concept tags and associated responses as input and trains the RNN to predict the future student response. An extension of DKT is the Deep Hierarchical Knowledge Tracing (DHKT) [21], which extended DKT to incorporate problem IDs in addition to concept tags.

However empirical experiments in [26, 23, 24] show that DKT does not appear to result in substantial improvement over many simpler models from classical IRT whose parameters and inferred states are psychologically meaningful. It is worth noting that the IRT variants considered in [26, 23, 24] use problem IDs as identifiers instead of skill IDs for DKT. Since multiple problem IDs can be tagged with the same skill IDs, we generally find that skill IDs repeat much more frequently than problem IDs. Thus, a comparison using skill IDs would likely be more favorable to a recurrent/sequential model like DKT. Of course, in considering only skill IDs we lose the ability to learn/exploit question-level information such as question difficulty. Moreover, producing skill IDs for each question requires substantial human effort and is often not feasible in practice. Furthermore all the experiments in [26, 23, 24] consider the 'New Student' evaluation protocol, which keeps a portion of the students as training sets and test on new students. Such an evaluation scenario may not be particularly meaningful in a real-world ITS and does not favor penalization models such as IRT, though online evaluation in [23, 24] mitigates such bias. Thus, the comparison study in [26, 23, 24] is not entirely satisfying and leaves open many questions regarding the potential benefits (or lack thereof) of deep RNNs for knowledge tracing.

Hybrid models. There are also several attempts to combine static models and sequential models to exploit advantages from both approaches, such as the FAST model in [5] and the LFKT model in [9]. In [10], these two approaches are compared and the experimental results show that these two hybrid models do not outperform a simple IRT model. The authors conjecture that the lack of improvement is due to a confounding between item identity and the question position in a (nearly deterministic) sequence of questions. In contrast to these more pessimistic results, in this paper we propose a hybrid model and show that it can harness the advantages from both static and sequential models in a way that outperforms both.

2.2 Session-based recommendation systems

A closely related application to knowledge tracing is that of predicting a user's preference for various items in a recommendation system. Among various recommendation systems, session based recommendation is the most closely related to knowledge tracing. For example, a session-based recommendation model, GRU4Rec, is proposed in [6] that has a similar architecture as DKT. However, GRU4Rec does not consider user identifications as inputs. An alternative approach – the Recurrent Recommender network (RRN) [25] – is capable of both modelling the seasonal evolution of items and tracking the user preferences over time. RRNs use a matrix factorization to model the stationary component of the user and item embeddings, and then two Long Short-Term Networks (LSTMs) to track the dynamic component of these embeddings.

Though similar, there are some notable differences between product recommendation and knowledge tracing. First, user preferences tend to change much more slowly compared to student knowledge. Second, student interactions with questions have a significant impact on student knowledge, while in contrast interactions with an item (watching a movie,

buying a product, etc.) typically have a mild impact at most on user preferences. Third, in a recommendation context, user responses may contain important implicit feedback [7]. For example, we can conclude that a user will watch a movie or buy a product because he/she likes it, even if the user does not give explicit feedback. However, students typically have limited freedom to choose which questions to answer. These differences have important algorithmic implications.

3. THE DYNEMB FRAMEWORK

3.1 System architecture

In this section we describe a novel framework for tracking student knowledge, dubbed *DynEmb*, that learns a *static* question embedding but exploits *sequential* models of the temporal dynamics of student-question interactions to track the knowledge states of the students. We will represent our training data as a sequence of interactions of the form $\mathcal{R}_t = (s_t, q_t, r_t, o_t)$. Each interaction \mathcal{R}_t involves a student s_t and a question q_t . We assume there are M questions and N students. The response to the question is denoted r_t , which is most commonly a correct/incorrect binary outcome or occasionally a numerical score. In this paper we focus mainly on the binary case, but the underlying framework can easily extend to the more general setting. Finally, we let o_t denote other information about the interaction that may be relevant, including – but not limited to – time stamps, questions tags, platform (e.g., paper, computer, mobile, etc.), and question text descriptions.

The goal of *DynEmb* is to predict student responses to future questions given a historical sequence of interactions $\{\mathcal{R}_i\}_{i=1}^n$. Specifically, given a new student-question pair (s_t, q_t) and any additional information o_t if available, our goal is to predict r_t . *DynEmb* has two main components, each of which are trained independently (see Figure 1). The first component *QuestionEmb* generates a d -dimensional *question embedding* $W_{q_t} \in \mathbb{R}^d$ from $\{\mathcal{R}_i\}_{i=1}^n$ using standard matrix factorization techniques described in more detail below. The second component *StudentDyn* learns to track each student’s knowledge state using a sequential model that takes the student’s past sequence of question embeddings $\{W_{q_i}\}_{i=1}^{t-1}$ and responses $\{r_i\}_{i=1}^{t-1}$ as inputs and produces a dynamic student embedding $Z_{s_t}(t) \in \mathbb{R}^d$. The sequential model could be a “vanilla” RNN, a long short-term memory (LSTM) network, a gated recurrent unit (GRU), a memory network with attention, or others. In this work we use an LSTM in the *StudentDyn* component by default. After obtaining the (static) question embedding W_{q_t} and the (dynamic) student embedding Z_{s_t} , the predicted probability of a correct response is computed via

$$\hat{r}_t = \phi(\langle W_{q_t}, Z_{s_t}(t) \rangle + b_{q_t}), \quad (1)$$

where b_{q_t} is a scalar that represents a bias learned for each question and ϕ is a sigmoid activation function. We describe these components in further detail below.

QuestionEmb. The *QuestionEmb* component uses an ℓ_2 -regularized biased matrix factorization model to learn a static latent embedding for the questions. More specifically, in this component we learn both a question embedding W and a student embedding Z , where $W \in \mathbb{R}^{N \times d}$ is a matrix whose

columns correspond to the question embedding vectors (the W_q ’s) and $Z \in \mathbb{R}^{M \times d}$ is a matrix whose columns correspond to the student embedding vectors (the Z_s ’s). These are learned via the following optimization problem:

$$\arg \min_{W, Z, b, c} \sum_{t=1}^n \mathcal{L}(r_t, \phi(\langle W_{q_t}, Z_{s_t} \rangle + b_{q_t} + c_{s_t})) + \lambda (\|W\|_F^2 + \|Z\|_F^2), \quad (2)$$

where b and c are vectors of question and student “biases” respectively, λ is the regularization parameter, and $\mathcal{L}(y, x) = -(y \log(x) + (1 - y) \log(1 - x))$ is the log loss function. This is inspired by the observations in [27] that if the question embedding W is static, then one can still use conventional matrix factorization to recover W , even though the other factors Z may actually be changing over time. Finally, we note that while (2) is a non-convex optimization problem, simple optimization algorithms exist that provably converge to a global minimum [8, 4].

StudentDyn. The *StudentDyn* component uses an RNN to sequentially generate a student embedding after each interaction. For the case of a binary response, r_{t-1} , the input to the recurrent neural network is the Kronecker product of the question embedding learned by the *QuestionEmb* component ($W_{q_{t-1}}$) and the vector $[r_{t-1}, 1 - r_{t-1}]^T$. At time step t , an interaction between student s_t and question q_t is predicted via the model in (1), and the RNN is trained to predict r_t . The dynamic student embedding $Z_{s_t}(t)$ is the internal hidden state of the RNN, which is then combined with W_{q_t} via (1) to obtain our final prediction.

3.2 Model training

To train *DynEmb*, we adopt a two-phase pretraining strategy. We first train the question embedding in the *QuestionEmb* component. We then feed the learned question embedding to the *StudentDyn* component to train the sequential model. Note that we keep the question embedding W and the biases b fixed when training the *StudentDyn* component. This embedding pretraining strategy not only speeds up the training process, but also produces better prediction performance compared to end-to-end training (see Section 4.4 for an experimental justification). Similar pretraining strategies are widely used in learning complex models (e.g., for machine translation [14] and sentiment analysis [17]).

Compared to DKT [13], DKVMN [28], and other sequential knowledge tracing models, the explicit question embedding learned directly from interactions based on matrix factorization seems to be more robust. In fact, in our experiments we have observed that if we replace the (frequently repeating) concept/skill tags in DKT and DKVMN with the (much less frequently repeating) question identifiers, then both DKT and DKVMN will have significant performance degradation and require intensive computational resources to train. However, our model can track student knowledge using the pretrained question embedding instead of concept/skill tags. This allows our approach to exploit question difficulty information and scales well, especially when concept/skill tags are not available.

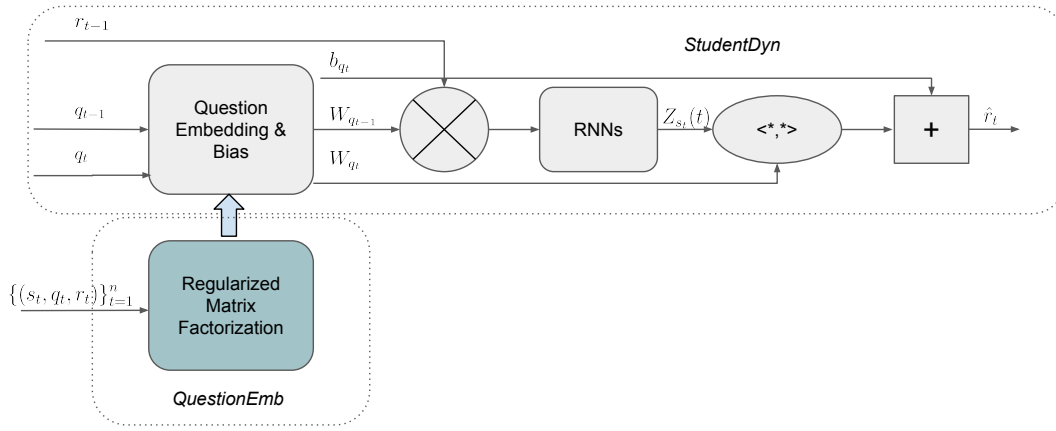


Figure 1: Architecture for *DynEmb*. First we train *QuestionEmb* to obtain question embedding W and bias b . Then we train the RNNs using past item embedding $W_{q_{t-1}}$ and response r_{t-1} as inputs to track student knowledge.

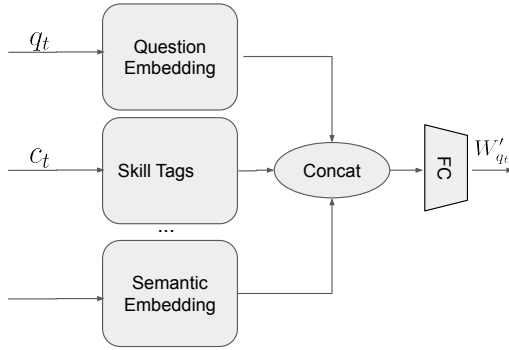


Figure 2: Multiple input fields.

3.3 Integrating skill tag information

If manually-labeled skill tag information is available for each question, then it is convenient and beneficial to incorporate this information into the *DynEmb* framework. However the question latent space learned via the matrix factorization might be different from the latent space constructed by manual labeling. One simple method to exploit both approaches consists of concatenating the two latent question embeddings to form a new latent question embedding. The skill tags can be one-hot encoded. To further exploit the hierarchical relationship between questions and skill tags, we initialize a question’s embedding by the one-hot encoding of its corresponding skill tag, and put an additional ℓ_1 regularization on the objective in (2) to promote sparsity.

To control the dimensionality of the latent space, the concatenated embedding is followed by a fully connected (FC) layer with ReLU activation. This kind of integration scheme can be found in [2] and also enables easy incorporation of additional embeddings/fields, e.g., semantic embedding from question text.

Finally, the *StudentDyn* component uses an RNN to sequentially generate a student embedding after each interaction using this modified question embedding just as before. See Figure 2 for additional details.

4. EXPERIMENTS

In this section, we experimentally validate the effectiveness of the proposed *DynEmb* model on two tasks: prediction of response correctness for existing students and prediction of response correctness for new students. By conducting experiments on several data sets each and comparing with the relevant baselines, we show that:

1. *DynEmb* outperforms DKT by up to 5.43% and 3.74% in predicting the next response in the ‘New User’ and ‘Most Recent’ evaluation settings respectively (see definition in Section 4.1);
2. The performance of *DynEmb* is stable with respect to the dimensionality of the item embedding;
3. The proposed embedding pretraining strategy is a key component of the success of the *DynEmb* approach.

4.1 Experimental setting

We consider the following baselines:

- Algorithms that compute a static embedding: in this category, we compared with BMF [19]. We compare to both offline and online BMF.
- Knowledge tracing based on RNNs: we compare with the state-of-the-art DKT algorithm [13].

We report the Area Under the ROC Curve (AUC) for comparing the predicted probabilities of correctness for each response. AUC is threshold agnostic, and is widely used in the knowledge tracing literature.

We use two evaluation methods. The first is online response prediction for new users [13, 23]. In this setting, students are first split into training and testing populations. Each model is first trained on the training population. Then for each time $t > 1$ in each testing student’s history, we train the student-level parameters in the model on a new student, including both the training population and the first $t - 1$ interactions of the student history, computing the probability

that the t^{th} response is correct. In practice, we find that re-training and testing after each response is not computationally feasible for large datasets, in which case we perform online response prediction in batches. We denote this evaluation method the ‘New User’ setting. Our second method is to consider online response prediction for the the most recent interactions as in [23]. The procedure here, denoted the ‘Most Recent’ setting, is the same as in the ‘New User’ setting except that we consider only the most recent interactions for our testing population as the testing data set.

4.2 Experiment 1: Future response prediction

In this experiment, the task is to predict students’ response. The prediction task is: given all interactions up to time t , given the student s and question q involved in the interaction at time t , what is student s ’s response (correct/incorrect) to question q ?

We use the following data sets to evaluate performance on this task.

ASSISTments. This data set was gathered from ASSISTments’s skill builder problem sets, where students learn by working on similar questions until they can respond correctly n (usually 3) times in a row [11]. We use two one the provided data sets, “ASSISTment09” and “ASSISTment12.” Note that the authors updated “ASSISTment09” in 2017 (first found in [26]).

Cognitive Tutor. In the 2010 KDD Cup Challenge, the PSLC DataShop released several data sets from Carnegie Learning’s Cognitive Tutor in (Pre-)Algebra from the years 2005-2009 [18]. We use three of the “Development” data sets, “Algebra I 2005-2006,” “Algebra I 2006-2007,” and “Bridge to Algebra I 2006-2007.”

Preprocessing of data sets. As noted in [23], there are multiple records duplicating a single interaction (represented by a unique *order.id* value) in “ASSISTment09.” These duplicate rows arise when a single interaction is aligned with multiple skills. This provides DKT models access to the ground truth when making their predictions, which can artificially boost prediction results by a significant amount. We adopt two strategies to clean the data. The first is to discard rows duplicating a single interaction (as in [23]); the second is to combine these duplicating rows into a single row with a new skill tag as suggested by [26]. In this paper we removed duplicate and multiple-skill repeated records in all data sets to ensure fairness for the purpose of comparison. We also removed “not original” records as suggested by [26]. We do similar cleaning operation on the other data set “ASSISTment12”. For the Cognitive Tutor data sets, we form problem identifiers from the concatenation of the “Problem Name” and “Step Name” fields.

Implementation details. The dimensionality of the input to the RNNs in *DynEmb* is fixed at 100. The ℓ_2 regularization parameter in the *QuestionEmb* component is chosen using cross-validation based on standard BMF. The hyperparameters in the *StudentDyn* component are the same as DKT and chosen by cross-validation.

Results. Table 2 compares the results of *DynEmb* with the baseline. We observe that *DynEmb* significantly outperforms the best baseline in all datasets in terms of AUC on the three datasets up to 5.43%.

4.3 Experiment 2: Robustness to embedding dimensionality

In this section, we study the effect of the dynamic embedding dimensionality on the tracking performance. In this study we use the “ASSISTment09” and Cognitive Tutor “Algebra I 2005” (“CT05” for short) datasets, which have the smallest number of interactions from the two tutoring systems respectively. The effect on other datasets is similar and omitted for the sake of brevity. We will test on the response prediction task. As we can see from Figure 3, the performance by AUC of *DynEmb* is quite stable over a wide range of embedding dimensionalities. This robustness is an additional attractive feature of our approach.

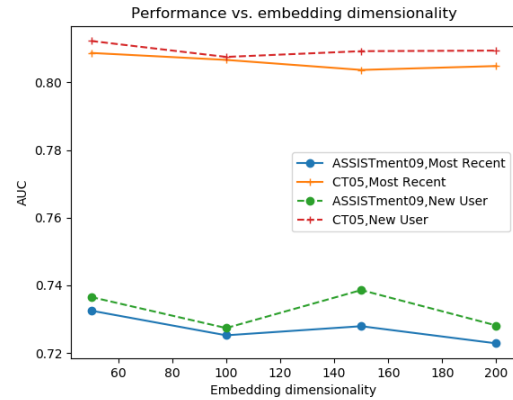


Figure 3: Performance versus embedding dimensionality.

4.4 Experiment 3: Embedding pretraining vs. end-to-end training

In this section we demonstrate why *DynEmb* uses pretraining for the question embedding. The dataset used in this section is “ASSISTment09.” We use the “Most Recent” evaluation method. In Figure 4, we can see that end-to-end (E2E for short) training (with/without pretraining the question embedding) will cause over-fitting, while the learning curve of proposed pretraining strategy does not suffer from over-fitting or under-fitting. Of course, another advantage of pretraining is its improved computational efficiency. The combination of these two factors provides powerful evidence for choosing pretraining over an end-to-end training strategy in this framework.

4.5 Experiment 4: Visualizing question embedding

Though the latent space of the question embedding learned via matrix factorization is not explicitly aligned with the latent space formed by the manually-labeled skill tags that were provided, the proposed question embedding initialization and sparsity promotion is remarkably effective at aligning the question embedding space with the manually constructed skill embedding space. This provides additional se-

Table 1: Overview of data sets.

Data set	Number of				Ratio of correctness	Description
	Skills	Problems	Students	Responses		
ASSISTments	101	13111	4003	214424	0.658	2009
	265	47124	28998	2623624	0.699	2012
Cognitive Tutor	90	210710	574	809693	0.767	Algebra I 2005
	488	580531	1338	2270384	0.772	Algebra I 2006
	494	207856	1146	3679188	0.888	Bridge to Algebra 2006

Table 2: Future response prediction experiment: Table comparing the performance of *DynEmb* (concatenating question and skill embedding) with baselines, in terms of AUC. *DynEmb* outperforms the best baseline by up to 5.43%. We also list the performance of *DynEmb* with only question embeddings.

Evaluation method	Model	BMF		DKT	DynEmb		Improvement
		offline	online		Question	Concat	
New User	ASSISTment09	0.67	0.686	0.727	0.725	0.739	1.65%
	ASSISTment12	0.694	0.717	0.709	0.722	0.736	2.65%
	Algebra I 2005	0.761	0.763	0.773	0.803	0.815	5.43%
	Algebra I 2006	0.761	0.786	0.808	0.805	0.821	1.61%
	Bridge to Algebra 2006	0.838	0.844	0.856	0.868	0.873	1.99%
Most Recent	ASSISTment09	0.706	0.727	0.661	0.738	0.727	0.00%
	ASSISTment12	0.67	0.696	0.71	0.692	0.714	0.56%
	Algebra I 2005	0.744	0.763	0.779	0.791	0.808	3.72%
	Algebra I 2006	0.761	0.782	0.801	0.813	0.822	2.62%
	Bridge to Algebra 2006	0.831	0.839	0.847	0.859	0.865	2.13%

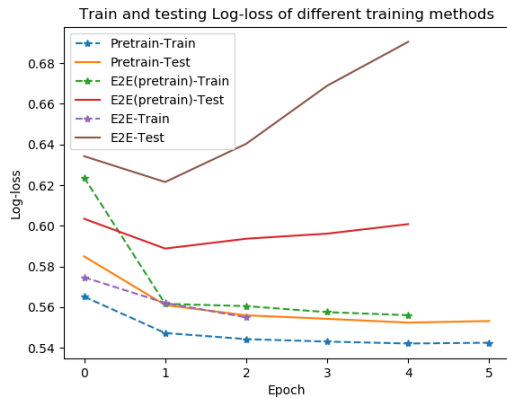


Figure 4: Training and testing log-loss of different training methods.

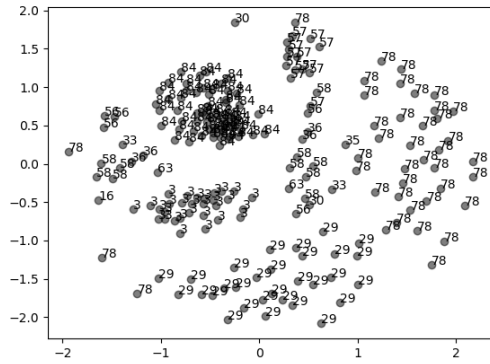


Figure 5: Visualization of the embedding of random selection of 200 questions by multidimensional scaling.

mantic meaning for the learned question embedding, which improves model interpretability. Figure 5 shows clear clustering of question embedding with respect to the associated skills (indicated by skill identifiers).

5. CONCLUSION AND DISCUSSION

In this paper we presented a framework to track student knowledge in an ITS by utilizing techniques from matrix factorization/embedding and RNNs. Our framework can track student knowledge without the concept/skill tag information required by other knowledge tracing models, e.g., DKT [13] and its variants. This avoids labor-intensive manual tagging. Taking advantage of additional latent question embeddings, our framework outperforms recent state of the art knowledge tracing models using RNNs. By constructing an embedding of the questions via matrix factorization in addition to skill tags, our framework can fuse question-level

and skill-level information. The *DynEmb* framework is also flexible in that it can accommodate various matrix factorization techniques and dynamical models, which makes it a promising avenue for future research and development of algorithms for knowledge tracing.

However, in the context of a real-world implementation, several challenges remain regarding how to design a practical *DynEmb* based system for knowledge tracing. For example, developing a method amenable to deployment in an online setting will require additional algorithmic improvement. Another challenge concerns how to incorporate additional sources of auxiliary information not considered here, such as question text or details about additional student interactions with an ITS (browsing history, textbook interactions, etc.) to best exploit all of the information that might be available. We believe that the *DynEmb* framework provides a natural platform to address such challenges.

6. REFERENCES

- [1] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis—a general method for cognitive model evaluation and improvement. In *International Conference on Intelligent Tutoring Systems*, pages 164–175. Springer, 2006.
- [2] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10. ACM, 2016.
- [3] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.
- [4] R. Ge, J. D. Lee, and T. Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.
- [5] J. González-Brenes, Y. Huang, and P. Brusilovsky. General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In *The 7th International Conference on Educational Data Mining*, pages 84–91. University of Pittsburgh, 2014.
- [6] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.
- [7] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. Ieee, 2008.
- [8] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.
- [9] M. Khajah, R. Wing, R. Lindsey, and M. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *Educational Data Mining 2014*. Citeseer, 2014.
- [10] M. M. Khajah, Y. Huang, J. P. González-Brenes, M. C. Mozer, and P. Brusilovsky. Integrating knowledge tracing and item response theory: A tale of two frameworks. In *CEUR Workshop Proceedings*, volume 1181, pages 7–15. University of Pittsburgh, 2014.
- [11] Z. Pardos. Assistments dataset homepage. <https://sites.google.com/site/assistmentsdata/home/>.
- [12] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Performance factors analysis—a new alternative to knowledge tracing. *Online Submission*, 2009.
- [13] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in neural information processing systems*, pages 505–513, 2015.
- [14] Y. Qi, D. S. Sachan, M. Felix, S. J. Padmanabhan, and G. Neubig. When and why are pre-trained word embeddings useful for neural machine translation? *arXiv preprint arXiv:1804.06323*, 2018.
- [15] G. Rasch. *Probabilistic Models for Some Intelligence and Attainment Tests*. Danish Institute for Educational Research, Copenhagen, Denmark, 1960.
- [16] S. Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.
- [17] S. M. Rezaeina, A. Ghodsi, and R. Rahmani. Improving the accuracy of pre-trained word embeddings for sentiment analysis. *arXiv preprint arXiv:1711.08609*, 2017.
- [18] J. Stamper, A. Niculescu-mizil, S. Ritter, G. G.J Gordon, and K. Koedinger. Challenged data sets from kdd cup 2010. <https://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp>.
- [19] N. Thai-Nghe, L. Drumond, A. Krohn-Grimberghe, and L. Schmidt-Thieme. Recommender system for predicting student performance. *Procedia Computer Science*, 1(2):2811–2819, 2010.
- [20] W. van der Linden and R. Hambleton, editors. *Handbook of Modern Item Reponse Theory*. Springer-Verlag, New York, NY, 2010.
- [21] T. Wang, F. Ma, and J. Gao. Deep hierarchical knowledge tracing. In *The 12th International Conference on Educational Data Mining*, pages 671–674. University of Buffalo, 2019.
- [22] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.
- [23] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham. Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation. *arXiv preprint arXiv:1604.02336*, 2016.
- [24] K. H. Wilson, X. Xiong, M. Khajah, R. V. Lindsey, S. Zhao, Y. Karklin, E. G. Van Inwegen, B. Han, C. Ekanadham, J. E. Beck, et al. Estimating student proficiency: Deep learning is not the panacea. In *In Neural Information Processing Systems, Workshop on Machine Learning for Education*, page 3, 2016.
- [25] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 495–503. ACM, 2017.
- [26] X. Xiong, S. Zhao, E. G. Van Inwegen, and J. E. Beck. Going deeper with deep knowledge tracing. *International Educational Data Mining Society*, 2016.
- [27] L. Xu and M. A. Davenport. Simultaneous recovery of a series of low-rank matrices by locally weighted matrix smoothing. In *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 1–5. IEEE, 2017.
- [28] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pages 765–774. International World Wide Web Conferences Steering Committee, 2017.

Incorporating Task-specific Features into Deep Models to Classify Argument Components

Linting Xue
North Carolina State
University
Raleigh, North Carolina, USA
lxue3@ncsu.edu

Collin F. Lynch
North Carolina State
University
Raleigh, North Carolina, USA
cflynch@ncsu.edu

ABSTRACT

In order to effectively grade persuasive writing we must be able to reliably identify and extract argument structures. In order to do this we must classify arguments by their structural roles (e.g., major claim, claim, and premise). Current approaches to classification typically rely on statistical models with heavy feature-engineering or on deep neural-networks that do not consider prior knowledge or other secondary features. Little research has been carried out to investigate if we can incorporate features into deep models to address AM tasks. In this work, we propose to incorporate lightweight features into deep models to classify argument components. We experimented with two state of the art (SOTA) approaches: 1) linear-Long-Short-Term Memory (LSTM) models with concatenated feature vectors; or 2) Directed Acyclic Graph (DAG) structured LSTMs. In our models we incorporated the features of argument position (e.g., if the argument is in the first paragraph) and prior knowledge of discourse indicators (e.g., in conclusion, for example). We use two baselines in our work: 1) prior work using SVM models with heavy feature engineering; 2) traditional linear-Bi-LSTMs with no task-specific features.

Our results show that with a comparatively small number of lightweight features, both linear-Bi-LSTMs and DAG-Bi-LSTMs outperform SVM models that depend on more heavy feature engineering, and outperform linear-Bi-LSTMs with only general word embedding features. These results suggest that incorporating task-specific elements into deep models may potentially benefit argument mining tasks.

Keywords

Argument component classification, deep learning, feature engineering, DAG-LSTMs, LSTMs, argument structures, argumentation mining, automated essay grading

1. INTRODUCTION

Current automated essay grading systems are typically focused on the syntactic and semantic analysis of written arguments via Natural Language Processing (NLP) techniques (as in [7, 23, 3]). These

systems are typically designed to evaluate arguments on the basis of: general readability (e.g., the number of prepositions and relative pronouns or the complexity of the sentence structure); shallow semantic analysis (e.g., lexical semantics or the analysis of the relationship among named entities); and syntax analysis (e.g., grammatical analysis). To the extent that argument structures considered in this work have been focused on the limited identification of individual components (e.g., hypothesis statements [4]), or on manual analysis by human experts [14], which is costly and time-consuming. Few existing systems perform any automatic analysis of the argumentative structures or seek to identify structural flaws due to the lack of an auto-extraction mechanism in the system.

In order to parse arguments it is necessary to extract the basic components. Extracting argument structures (EAS) is one of the essential tasks of argumentation mining (AM). EAS can be divided into three sub-tasks: 1) **argument component identification (ACI)** breaking down the text into argument units; 2) **argument component classification (ACC)** of classifying argument component (ACs) into types; 3) **argument relation identification (ARI)** of identifying the relationships between each pair of ACs. Prior researchers have focused on different subsets of these tasks (e.g., [27] addressed ACI, ACC, and ARI separately, [24] jointly modeled ACI and ACC) or built end-to-end models that address them sequentially (e.g. [18]). Our goal in this work, by contrast, is to investigate how to incorporate task-specific features into deep learning models and whether those features can improve our models' performance on the task of ACC. We carried out our work using an argumentation schema developed by Stab and Gurevych on a corpus of 402 persuasive essays (PE) [27]. As part of this work, we replicated their work on ACC and used it as a baseline model.

Most current approaches to ACC either rely on heavy feature engineering [27, 16, 22, 12] or use deep models that only consider pre-trained word embeddings with no other secondary features [19, 24, 11]. Little research to date has been focused on incorporating prior knowledge or lightweight features into deep models for AM tasks. To the best of our knowledge, Lugini and Litman carried out the only work that adds features into LSTM based models to address ACC on the argument dataset of classroom discussions [13]. In that work, they considered a set of features including semantic-density features (e.g. the number of pronouns), lexical features (e.g., uni-gram and bi-gram), and syntactic features from speech tags. They combined the feature matrix and LSTMs hidden output for classification. They showed that the features boosted the deep model performance.

In our work, we investigated whether or not it is possible to incor-

Linting Xue and Collin Lynch "Incorporating Task-specific Features into Deep Models to Classify Argument Components" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 531 - 537

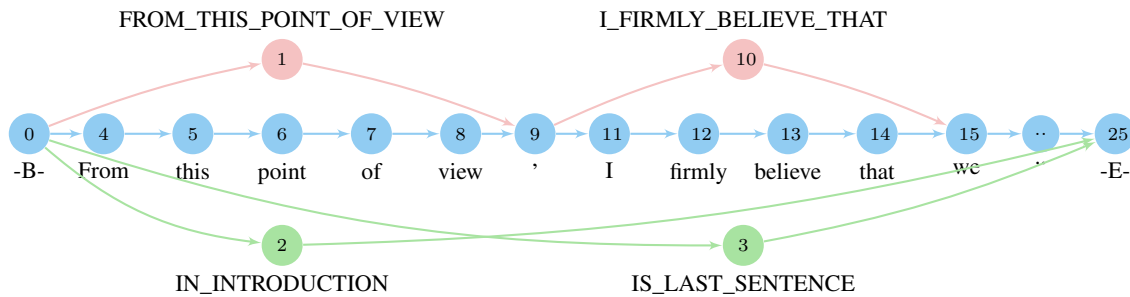


Figure 1: An example of DAG LSTM modeling an AC.

porate lightweight features into deep models to address ACC on PE dataset with different feature sets and deep models. In this work we considered prior research on the prior knowledge of discourse indicators and position features. The discourse indicators have been shown are potential features for identifying the argumentative section of online product reviews [30]. Researchers have also demonstrated that the structural features of AC position (e.g., if the AC shows in the introduction, if an AC is the first sentence of one paragraph) and token statistics (e.g., number of tokens of an AC) are the most effective features for AC classification [27]. For this study we only considered the position information for ACs. We chose to focus on these two features because these two are the most informative and also require the least amount of feature-engineering.

For our deep models, we experimented with Bi-LSTMs. We encoded the incorporated features by one-hot encoding, computed the element-wise summation of feature vectors, and then combined feature vectors with Bi-LSTMs output for prediction. This approach is the same as the work in [13]. We also considered bidirectional DAG-Structured Recurrent Neural Networks (RNNs) to incorporate features. **DAG-RNNs**, also known as Neural Lattice Language models, are an extension of linear-chained RNN models that can consume DAG-structured input [32]. If we treat the text as a linear path, the prior knowledge and secondary features can be added as new edges on the path to form a DAG structure. The discourse features are connected to the parent- and child- nodes of the related tokens, similar to the work of [32]. For position features, we simply connected them with the two special sequence delimiters which indicate the beginning and end of the sentences. Figure 1 shows an example of DAG input with discourse indicators of “FROM_THIS_POINT_OF_VIEW” and “I_FIRMLY_BELIEVE_THAT” in red and position features of “IN_INTRODUCTION” and “IS_LAST_SENTENCE” in green. The original input is in the blue nodes. Token “-B-” and “-E-” are special sequence delimiters. The nodes are indexed in topological order, as this is the order in which the one-directional DAG model consumes the input sequence. For bidirectional DAG models, we simply reverse the order. The intuition behind this approach is that it mimics how humans read and annotate essays as humans can incorporate linguistic intuition to determine the role of the ACs in written argumentation. For example, if a sentence appears to be the last sentence of the introduction in a five-paragraph essay, it most likely contains the author’s standpoint, i.e., claim.

DAG-RNNs have been used to incorporate linguistic knowledge (e.g., the non-compositional phrase in the form of n-gram) for sentiment classification [32]. They have also achieved SOTA results in many other NLP tasks such as neural machine translation [28], speech translation [25], and language modeling [2]. In this work,

we utilized LSTMs, a special kind of RNN and building off Zhu et al.’s work [32], we implemented a DAG-LSTM in Tensorflow with a different hidden state bagging function (discussed in section 4).

Our results show that linear-Bi-LSTMs with no task-specific features performed worse than traditional models. However once we incorporated our two features, both the linear-Bi-LSTMs and DAG-Bi-LSTMs outperform general Bi-LSTMs with no features and they outperform other models that rely on heavy feature-engineering. DAG-Bi-LSTMs slightly outperform the linear-Bi-LSTMs when considering both features. The linear-Bi-LSTMs with only position features yield the best results.

The significance of this work is as follows. 1) Our work serves as the basis for automated essay grading systems, and can be applied to extract argument structures for detecting structural flaws. 2) We addressed a common issue in NLP that deep models tend to yield lower performance on small datasets. We showed that deep models can benefit from lightweight features and yield better performance. 3) We experimented with DAG-LSTMs to incorporate features on text classification tasks. We showed that it could be a promising architecture to incorporate features into sequence models. 4) We tested the same approach used in [13] to combine features with LSTMs on a different dataset. Our results are consistent with their work.

2. RELATED WORK

2.1 AC Classification

Most of the prior work on AC classification relies on traditional classification models and heavy feature engineering. In [27], researchers applied multiclass SVMs to classify ACs using structural, lexical, syntactic, discourse indicator, and contextual features. They obtained an F1 score of 0.794 on the PE corpus. In [26] and [18], authors performed classification task on a small portion of the PE dataset, again relying on extracted features. In [10], Namhee et al. analyzed online comments to identify and classify subjective claims using lexical and syntactic features. In [16], researchers worked to classify ACs on legal documents using extracted features while Niall et al. applied kernel methods for argument detection and classification on AraucariaDB dataset [22]. Different from the above, we only considered prior knowledge of discourse indicators and structural information of position features.

Many researchers have begun to explore the application of deep neural-network models to argument mining. In [11], authors experimented with CNN and RNN models to detect the claims [1] and evidences [21] on Wikipedia datasets. Potash et al. proposed a joint sequence-to-sequence model with attention to predict the links between ACs and classify ACs on the PE dataset, where they consid-

ered the sequential nature of ACs [19]. In our work, by contrast, we focused on incorporating prior knowledge to deep neural-networks to classify the ACs alone.

To the best of our knowledge, research from [13] is the only work that combines feature engineering and deep models to address ACC on classroom discussion. They showed that SOTA deep models with only pre-trained embeddings performed poorly on their dataset. However, by including secondary features they improved the performance substantially. The features included: semantic-density features (e.g., number of pro-nouns, descriptive word-level statistics, number of occurrences of words of different lengths), lexical features (e.g., tf-idf feature for each unigram and bi-gram, descriptive argument move-level statistics), and syntactic features (e.g., unigrams, bigrams, and tri-grams of part of speech tags). In their work, they experimented with Convolutional neural network models and LSTMs. Their results showed that the model’s performance was improved after adding the secondary features. In our work, we considered the same approach to incorporate features into linear-LSTMs and compare the model performance with DAG-LSTMs.

2.2 DAG-RNNs

DAG-RNNs, also known as Neural Lattice Language (NLL) models, are extensions of chain-structured RNNs [32, 28]. These models, first proposed by Zhu et al. in [32], leverage DAG structures to incorporate external semantics such as n-gram sentiment tags and expert annotations to improve performance on sentiment classification. Su et al. introduced NLL-based Gated Recurrent Units (GRUs) to encoder multiple word segmentation of Chinese text for translation [28]. Sperber et al. later used NLL-based GRU models to consume word lattices from the up-stream modes of the speech recognizer for speech translation [25]. These lattices were annotated with posterior probabilities on the alternative translation paths. And finally, in [2], researchers demonstrated that the NLL models outperformed the LSTM-based models at the task of language modeling when incorporating multi-word phrases (n-grams) and multiple-embeddings for polysemy. However, little research has been done to utilize DAG-RNNs to integrate features for AM tasks.

3. DATASET

The PE dataset was developed by [27]. It contains 402 essays from the online community *essayforum*¹. The forum provides writing feedback for different kinds of text. Students can post practice essays for standardized tests in the community and obtain feedback about their writing skills. The dataset was randomly selected from the *writing feedback* section of the forum. The dataset comes with three argument components: *major claim* indicating the author’s standpoint on the given controversial topic; *claim* of sub-standpoints that supports (“for”) or attacks (“against”) the major claim; *premise* that is the reason of the argument which supports or attacks the claim. Table 1 shows the class distribution of the PE corpus. The average number of tokens in *major claims*, *claims*, and *premises* are 19, 23 and 21, respectively.

	Major Claim	Claim	Premise
Train & Dev	598	1202	3023
Test	153	304	809

Table 1: Number of instance in each class

¹<https://essayforum.com/>

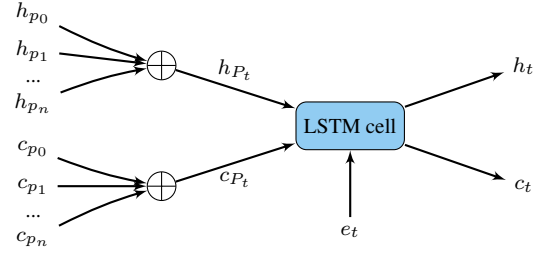


Figure 2: An unit of DAG-LSTM.

4. METHODS

4.1 Linear-Bi-LSTMs

In traditional Bi-LSTM models, ACs are encoded using Glove embeddings [17] that are obtained from training on large Wikipedia datasets. The encoded ACs are fed to Bi-LSTMs, and the last hidden states are passed to the softmax layer for prediction. To incorporate features, we used a one-hot vector to represent each feature and summed up the features vectors that are related to an AC. For example, if we have a total of three features in the feature space that are “if an AC is in the introduction”, “if an AC is in the conclusion”, and if an AC contains discourse indicator of “in conclusion”. We use one-hot vectors to represent three features as [0, 0, 1], [0, 1, 0], and [1, 0, 0], respectively. When we have one example AC that contains a discourse indicator of “in conclusion,” and this AC is in conclusion paragraph, we sum up two feature vectors by elements to get a vector of [0, 1, 1], which represents the feature for this AC. Then we concatenate the vectors on the hidden output of Bi-LSTMs for final prediction. The same approach has been used in [13]

4.2 DAG-Bi-LSTMs

We implemented the DAG-Bi-LSTMs using the TensorFlow platform.¹ The DAG-Bi-LSTMs in our work is similar to the models described in [32]. However, we applied a different hidden state merge function. While Zhu et al. used binarization, we elected to sum the parent hidden states as suggested in the TreeLSTM work [29]. Intuitively, by summing the previous states, we expect the DAG-models to learn both the summarized linear history and the incorporated knowledge. We used the same one-hot method to encode the incorporated features.

For sequential inputs the linear-LSTM models calculate hidden states h_t and cell states c_t based upon the proceeding hidden state h_{t-1} , cell states c_{t-1} and the input embedding e_t of token x_t as:

$$h_t, c_t = LSTM(h_{t-1}, c_{t-1}, e_t, \theta) \quad (1)$$

The primary difference between DAG- and linear- LSTMs is that the former can have multiple parent and child states, as shown in Figure 2, while the latter cannot. Given a DAG input, h_{p_i} indicates a set of parent states at t time step, where $i = 0, 1, \dots, or n$ and $p_i \in P$. The DAG model first gathers its parent hidden states h_{p_i} and then sums over the parents’ hidden states and over the parents’ cell states as follows:

$$h_{P_t} = \sum_{p_i \in P_t} h_{p_i} \quad c_{P_t} = \sum_{p_i \in P_t} c_{p_i} \quad (2)$$

¹We also experimented with DAG-Bi-Gated Recurrent Units, but DAG-Bi-LSTMs yielded better results.

The remainder of the DAG process is similar to that of linear-Bi-LSTMs in that h_{P_t}, c_{P_t}, e_t are fed to standard LSTM unit to generate new hidden, cell states of h_t, c_t , which are then copied to the child states. Finally, the last hidden states are fed to Multi-Layered Perceptrons (MLPs) for prediction.

4.3 Prior Knowledge and Features

In this work, we considered the prior knowledge of discourse indicators and the AC position features. In prior work [27], Stab and Gurevych collected a list of hand-crafted features for ACC tasks, including lexical, structural, discourse indicator, contextual, syntactic, etc. The detailed explanations of the features can be found in Section 5.3.1 of [27]. For discourse indicators, they include five categories: *forward indicators* (e.g., “therefore”); *backward indicators* (e.g., “because”); *thesis indicators* (e.g., “in my opinion”); *rebuttal indicators* (e.g., “although”); and *first-person indicators* (e.g., “I”, “me”). For the position features, we annotated sentences if they were the first/last sentence and if they showed up in the last/first paragraph. The annotations are in the *special* n-gram form (e.g., “IS_LAST_SENTENCE”, “_THEREFORE_”) so that they are distinguished from original corpus. We also experimented with annotating the discourse indicators by category, such that the forward indicators were annotated as “FORWARD_INDICATORS”.

5. EXPERIMENTAL SETUP

We carried out a series of experiments using the same static training/testing split as in prior work [27]. Since the corpus does not have a designated development set, we used stratified sampling to select 15% of the training set to tune our hyper-parameters and reported our final results on the designated test set. We ran each experiment five times and reported the average Macro-F1 score of the test dataset.

We carried out four distinct experiments: **Base-SVMs**, which replicates the work in [27] with multiclass SVMs using polynomial kernels on a set of features; **Base-LSTMs**, which are baseline models of general Bi-LSTMs with no secondary features; **LSTMs** and **DAG-LSTMs** refers to Bi-LSTMs and DAG-Bi-LSTMs with task-specific features.

We used a grid search for hyper-parameter tuning, and we used the same set of parameters across all the models. We used 300-dimensional GloVe embeddings [17]. Tokens not present in the pre-trained embeddings or not features were randomly initialized with uniform samples from range $[-\sqrt{\frac{3}{dim}}, \sqrt{\frac{3}{dim}}]$ [15] where dim is the dimension of the embeddings 300. All of the tokens in the test and dev sets but not in the training set have one unique random embedding. The embeddings were fixed during training. We then used the Adam optimization algorithm [9] with a learning rate of 0.005, a batch size of 32, a layer LSTM with a hidden size of 64, and a drop out rate of 0.2, and a layer tanh-MLP with a hidden size of 64.

6. RESULTS & DISCUSSION

In this section, we will discuss the overall results. Later we will talk about how each feature impacts the linear-LSTMs and DAG-LSTMs by comparing them with traditional SVMs and linear-LSTMs with no task-specific features. In the end, we will compare the performance of linear-LSTMs and DAG-LSTMs.

6.1 Overall Results

Table 2 shows the results of each experiment and our baseline metrics. The standard deviations of the deep model results are all less than 0.009 over the runs. The first three columns show our benchmark, Stab and Gurevych’s results with SVMs on: all features, structural features alone (including AC position in the document and token statistics), and contextual features alone (including discourse indicators and the number of noun and verb phrases in an AC). The next two columns show two of our baseline models: Base-SVMs and Base-LSTMs. Then the rest shows the linear-LSTMs and DAG-LSTMs with the two features together and separately. *Pos* includes the position features are considered, while *dis* refers to the discourse indicators were incorporated, and *Pos-dis* indicates that both features are used.

Overall, for linear-LSTMs with only position features return the best macro-F1 score of 0.805 across the board. DAG-LSTMs with both position and discourse features return a very close score of 0.802. Drilling won, linear-LSTMs with position features also yield the best F1 score for claim and premise components, especially for claim components, the F1 score is increased by 23% over the base-LSTMs and increased by 4.5% over base-SVMs. For major claims, the SMVs from prior work still have the best F1 score.

Thus for traditional models with heavy feature engineering, our Base-SVMs are close to Stab and Gurevych’s result (SVMs) but with a lower F1 score on major claims. This may be due to minor differences in our feature extraction or the different experimental setting. Among three deep models, the base-LSTMs with only pre-trained word embeddings perform very poorly, and all the F1 scores are much lower than the SVM models, which is not very surprising because of the small amount of data. The trained models do not generalize well on test data. This may also be due to the fact that the pre-trained embeddings are obtained from training models on a large corpus of Wikipedia data [17], which can be thought of prior knowledge. However, Wikipedia is very different from the PEs, the writing is generally more formal; it is a product of collaborative work; and is heavily edited. PEs are most likely composed by non-native English writers. Thus the base-LSTMs with glove embeddings may not able to catch the semantic meaning in the PEs.

However, once we incorporated the position and discourse features, we obtained a very high Macro F1 score. We have F1 of 0.801 and 0.802 for linear and DAG models, a 16% improvement over Base-LSTMs with no features (0.633). These results imply that both models can utilize task-specific features to boost performance. When we compare deep models with heavy feature engineering based SVMs, we find that the deep models with only two features outperformed the SVMs with heavily features Engineering, which suggests that combining little features with deep models can improve the learning on AM tasks.

6.2 Position Features

When adding position features, both the linear models and DAG models outperformed the SVMs and base-LSTMs. The linear models return the best results. In fact, after incorporating position features, the major claims that were previously misclassified as premises by the Base-LSTMs were now either classified as claims or major claims in the linear and DAG models. However, while adding the position features improves the tasks on identifying the claims and major claims, they are not good at distinguishing the two. Looking deeper, the resulting models tend to be biased towards the major claim classification, especially for claims that show up in the introduction or conclusion. One possible explanation for this is that

	SVMs (Stab et al.)			Base-SVMs	Base-LSTMs	LSTMs			DAG-LSTMs		
Features	All	Pos	Dis	All	None	Pos-Dis	Pos	Dis	Pos-Dis	Pos	Dis
Major claim	.891	.803	.656	.844	.625	.832	.823	.650	.852*	.845	.645
Claim	.611	.551	.248	.640	.455	.670	.685*	.428	.659	.668	.498
Premise	.879	.870	.836	.886	.819	.903	.907*	.820	.896	.886	.797
Macro-F1	.794	.741	.580	.790	.633	.801*	.805*	.633	.802	.799	.647

Table 2: Results on classifying ACs on PE corpus. “All” column refers to the eight type of features in [27]. “Pos” column indicates the models with position features. “Dis” columns shows the model results with discourse features. * means significant improvement over Linear-LSTMs with no features. Bold indicates the best result per row.

the position features dominate the feature space in the PE dataset; models pay too much attention to the position features and too little attention to the semantic context.

Our results are consistent with prior work which has suggested that position features play an important role in classifying ACs on the PE dataset. As shown in Table 2 with only position features, the traditional SVMs can reach a macro F1 score of 0.741. The utility of this feature is relatively intuitive. In five-paragraph essays, the major claims usually show up in the first or last paragraphs. And in our PE dataset, 70% of the major claims were either the last sentence of the introduction or in the first sentence of conclusions, while 67% of key subsidiary claims show up in the first or last sentence of the middle paragraphs.

Our results also suggested that we can incorporate non-semantic features into deep models to help the model learn, especially these non-semantic features can not be captured by the word embedding features.

6.3 Discourse Indicator features

Interestingly, the performance of linear models was not improved after adding the discourse features, and DAG model’s performance was improved a little. When we examine the data more closely we see that one possible reason is the current discourse indicator list provided in [27] does not cover all the cases in the PE. We identified more discourse indicators that are not included in the list, such as thesis indicators of “it is believed that”, “to summarise”, “in short”, “it is undeniable”, and “I admit that” and forward indicators of “based on the above discussion”. We will address this problem in our future work.

We also experimented with incorporating discourse features by category. We ran another set of experiments based on the same model parameters setting as above. Below Table 3 shows the results.

Features	LSTMs		DAG-LSTMs	
	Pos-Dis	Dis	Pos-Dis	Dis
Major Claim	.843*	.662*	.859*	.675*
Claim	.666	.475*	.659	.506*
Premise	.899	.817	.894	.794
Macro-F1	.803*	.651*	.804*	.659*

Table 3: Results for incorporating discourse indicators as annotation types for Linear-LSTMs and DAG-LSTMs.* means improvement over the results that show in Table 2.

After incorporating discourse indicator features by category, the model’s performance was improved. In Table 3, * indicates the improvement over the results from n-gram discourse features. Both the linear and DAG models yield slightly better results on major claim and claim components, especially the major claim. Deeply looking at the results, some major claims were misclassified as claims before, which were correctly predicted here. The reason could be that when we consider the discourse indicators by category, we only have five features. For each feature, we have sufficient training examples for it. In fact, the thesis, first-person, forward, rebuttal, and backward indicators show up 701, 1535, 968, 719, and 1769 in the total data. Thus, it is easier for the models to capture the discourse features. However, when we considered them in the n-gram form, we have more than 100 of them. The number of them shows up in the ACs are much less than above. And some of them only show up once in the entire data, such as thesis indicators of “all things considered” and backward indicator of “is due to the fact that”, which prohibits the models learning useful information.

Overall, the discourse indicator features did not improve the model’s performance massively, which indicates that deep models with pre-trained embeddings already capture the semantic information. Thus, adding discourse indicator features does not help as much as position features on PE dataset.

6.4 Linear-LSTMs vs DAG-LSTMs

When adding both classes of features, the linear and DAG models yielded similar results for their macro-F1 score. However, the linear models outperformed the DAG models with the position features alone, and the DAG models utilized the discourse indicator features better in both the Table 2 and Table 3. One possible explanation is that when we concatenated the feature vectors on the last hidden output of linear models, the models are not able to learn the interactions between the discourse indicators and surrounding words. But the DAG models can learn those interactions by merging the hidden annotation states with current hidden states, and the current states contain the semantic information from all previous tokens. For example, for the DAG-input shown in Figure 1, we use the index of the DAG input to refer the time step that the LSTM unit processes the hidden state. At time step 15 of the forward training, we first merge the hidden output of state 10 and state 14, and then pass the merged hidden state as the hidden input of state 15. In this way, the DAG models consider both semantic meaning of all previous tokens and the discourse feature of “I firmly believe that”, and pass that information to the next hidden state. These results imply that DAG-models tend to utilize the semantic features better as they can learn the interaction between the features and tokens. However, they might not perform very well when we consider non-semantic features, such as the position features used here. One possible ap-

proach to address this problem is to combine two proposed methods to incorporate features. We can use DAG models to incorporate discourse indicator features and then concatenate the one-hot position feature vectors on the final hidden states for prediction.

7. CONCLUSIONS

In this work, we experimented with two approaches to combine feature engineering with deep models: linear-Bi-LSTMs with feature vectors concatenated on the hidden output and DAG-Bi-LSTMs. Our results show that both deep models could benefit from task-specific features, as both of them outperformed traditional models with heavy feature engineering and deep models with no task-specific features. We also show that the linear models handle the position feature better, and that the DAG models utilize the semantic features better since the linear models can not learn the interaction between discourse features and tokens. And finally we show that the deep models benefit more from position features than discourse indicator features on the PE dataset. Our results imply that when we apply the deep learning models to classify ACs, we could consider utilizing some task-specific features to guide the model learning.

This work can serve as a basis for the development of structurally-aware support platforms for reading and writing. This can include automated essay grading systems that detect and evaluate structural deficiencies as well as writing tutors that scaffold the construction of coherent essays or identify structural issues. As discussed in Section 1, current automated grading systems suffer from the lack of reliable auto-extraction mechanisms with most still relying on traditional ML models that use heavy feature engineering to function. Such work is costly and time consuming to develop and may not always generalize to other essay types. Our work addresses this problem by showing that lightweight features and off the shelf methods can outperform those methods. At the same time our work also showed that while traditional machine learning models are costly and deep learning models are sensitive to small datasets, as discussed by [8, 31], this limitation too can be addressed through the use of lightweight feature work to guide the deep models. By addressing these two problems we have shown a path for developing robust argument detection mechanisms for automated educational platforms using novel deep learning approaches a path that can lead to substantive improvements for students and educators.

8. FUTURE WORK

These preliminary results serve as a basis for our ongoing research, in which we are building an end-to-end model with feature engineering to address all three sub-tasks for argument structure extraction. For that work we will frame this task as sequence tagging problems. We propose to use linear-LSTM and DAG-LSTM based models with task-specific features to address EAS. We estimate that incorporating the task-specific features into end-to-end models can improve the model's performance compared to the deep models based on general word embedding [6].

In future work, we will also consider experimenting these two approaches on different argumentation datasets, and compare the results with fine-tuning SOTA language models (e.g. BERT [5], T5 [20]).

9. REFERENCES

- [1] E. Aharoni, A. Polnarov, T. Lavee, D. Hershcovich, R. Levy, R. Rinott, D. Gutfreund, and N. Slonim. A benchmark dataset for automatic detection of claims and evidence in the

context of controversial topics. In *Proceedings of the first workshop on argumentation mining*, pages 64–68, 2014.

- [2] J. Buckman and G. Neubig. Neural lattice language models. *TACL*, 6:529–541, 2018.
- [3] J. Burstein, C. Leacock, and R. Swartz. Automated evaluation of essays and short answers. 2001.
- [4] J. Burstein, D. Marcu, and K. Knight. Finding the write stuff: Automatic identification of discourse structure in student essays. *IEEE Intelligent Systems*, 18(1):32–39, 2003.
- [5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [6] S. Eger, J. Daxenberger, and I. Gurevych. Neural end-to-end learning for computational argumentation mining. In R. Barzilay and M. Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 11–22. Association for Computational Linguistics, 2017.
- [7] M. A. Hearst. The debate on automated essay grading. *IEEE Intelligent Systems and their Applications*, 15(5):22–37, 2000.
- [8] N. A. Khayati and V. Rus. Bi-gru capsule networks for student answers assessment. In *2019 KDD Workshop on Deep Learning for Education (DL4Ed)*, 2019.
- [9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [10] N. Kwon, L. Zhou, E. Hovy, and S. W. Shulman. Identifying and classifying subjective claims. In *Proceedings of the 8th annual international conference on Digital government research: bridging disciplines & domains*, pages 76–81. Digital Government Society of North America, 2007.
- [11] A. Laha and V. Raykar. An empirical evaluation of various deep learning architectures for bi-sequence classification tasks. *arXiv preprint arXiv:1607.04853*, 2016.
- [12] M. Lippi and P. Torroni. Context-independent claim detection for argument mining. In *IJCAI*, volume 15, pages 185–191, 2015.
- [13] L. Lugini and D. J. Litman. Argument component classification for classroom discussions. In N. Slonim and R. Aharonov, editors, *Proceedings of the 5th Workshop on Argument Mining, ArgMining@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 57–67. Association for Computational Linguistics, 2018.
- [14] C. F. Lynch, K. D. Ashley, and M. Chi. Can diagrams predict essay grades? In S. Trausan-Matu, K. E. Boyer, M. E. Crosby, and K. Panourgia, editors, *ITS, Lecture Notes*, pages 260–265. Springer, 2014.
- [15] X. Ma and E. H. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.
- [16] R. Mochales and M.-F. Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22, 2011.
- [17] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [18] I. Persing and V. Ng. End-to-end argumentation mining in student essays. In *Proceedings of the 2016 Conference of the*

North American Chapter of the Association for Computational Linguistics: *Human Language Technologies*, pages 1384–1394, 2016.

- [19] P. Potash, A. Romanov, and A. Rumshisky. Here’s my point: Joint pointer architecture for argument mining. *arXiv preprint arXiv:1612.08994*, 2016.
- [20] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.
- [21] R. Rinott, L. Dankin, C. Alzate, M. M. Khapra, E. Aharoni, and N. Slonim. Show me your evidence—an automatic method for context dependent evidence detection. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 440–450, 2015.
- [22] N. Rooney, H. Wang, and F. Browne. Applying kernel methods to argumentation mining. In *FLAIRS Conference*, 2012.
- [23] L. M. Rudner and T. Liang. Automated essay scoring using bayes’ theorem. *The Journal of Technology, Learning and Assessment*, 1(2), 2002.
- [24] C. Schulz, S. Eger, J. Daxenberger, T. Kahse, and I. Gurevych. Multi-task learning for argumentation mining in low-resource settings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 35–41, 2018.
- [25] M. Sperber, G. Neubig, J. Niehues, and A. Waibel. Neural lattice-to-sequence models for uncertain inputs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1380–1389, 2017.
- [26] C. Stab and I. Gurevych. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, 2014.
- [27] C. Stab and I. Gurevych. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659, 2017.
- [28] J. Su, Z. Tan, D. Xiong, R. Ji, X. Shi, and Y. Liu. Lattice-based recurrent neural network encoders for neural machine translation. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3302–3308, 2017.
- [29] K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.
- [30] A. Z. Wyner, J. Schneider, K. Atkinson, and T. J. Bench-Capon. Semi-automated argumentative analysis of online product reviews. *COMMA*, 245:43–50, 2012.
- [31] Y. Xu and C. F. Lynch. What do you want? applying deep learning models to detect question topics in mooc forum posts? In *2019 KDD Workshop on Deep Learning for Education (DL4Ed)*, 2019.
- [32] X. Zhu, P. Sobhani, and H. Guo. Dag-structured long short-term memory for semantic compositionality. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 917–926, 2016.

A Quantitative Machine Learning Approach to Master Students Admission for Professional Institutions

Yijun Zhao
Computer and Information
Science Department
Fordham University

Bryan Lackaye
Computer Science
Department
Northeastern University

Jennifer G. Dy
Electrical and Computer
Engineering Department
Northeastern University

Carla E. Brodley
Khoury College of Computer
Sciences
Northeastern University

ABSTRACT

Accurately predicting which students are best suited for graduate programs is beneficial to both students and colleges. In this paper, we propose a quantitative machine learning approach to predict an applicant's potential performance in the graduate program. Our work is based on a real world dataset consisting of MS in CS students in the College of Computer and Information Science program at Northeastern University. We address two challenges associated with our task: subjectivity in the data due to change of admission committee membership from year to year and the shortage of training data. Our experimental results demonstrate an effective predictive model that could serve as a Focus of Attention (FOA) tool for an admission committee.

Keywords

support vector machine (SVM), semi-supervised learning, learning using privileged information (LUPI), multi-task learning, Educational Data Mining (EDM), business intelligence in education

1. INTRODUCTION

Master's education is the fastest growing and largest component of the graduate enterprise in the United States. According to the 2016 joint survey conducted by the CGS (Council of Graduate Schools) and ETS (Educational Testing Service) [4], first-time enrollment in U.S. graduate programs reached a record high total of 506,927 students in Fall 2015. Because of the rise in applicants, the admissions process may become increasingly tedious and challenging. The ETS has established standardized tests (such as the GRE) to help evaluate applicants' quantitative, reading, and writing skills, but these scores alone are far from indicative of success-

ful students. Although applicants' previous achievements can demonstrate excellence, students with high GPAs from prestigious universities do not always excel in their graduate studies.

In this paper, we take a quantitative machine learning approach to predict the outlook of applicants' graduate studies based on features extracted from their application materials. The training data for our model are empirically admitted students with their performance measures in the graduate program. In particular, we have a real world dataset from Northeastern University's MS in Computer Science (MSCS) program, consisting of MS students from 2009 to 2012. We use a student's overall GPA in the MSCS program as his/her performance measure. Our model aims to identify the top 20% and bottom 20% performing students respectively (see details in Section 4.1).

Two challenges arise when learning with this data. First, the data involves the admission committee's (possibly subjective) evaluation. Specifically, some members of the committee may be biased in weighing a particular set of standards (e.g., GRE scores), while others may be in favor of different measures. This issue is particularly acute when the admission committee/policy changes from year to year. As a result, it can be difficult to form an accurate predictor directly from the entire dataset. Another challenge is the limitation of the training data. We have a total of 454 labeled training samples (all admitted students) from 2009 to 2012. On the other hand, we have over 2000 applications that are either rejected (i.e., not admitted) or declined (i.e., admitted but not enrolled), which can serve as an unlabeled auxiliary dataset. Our conjecture is that building a semi-supervised model leveraging the large set of unlabeled data may lead to a superior performance compared to using the labeled data alone.

Our model is inspired by two existing frameworks: SVM+ [12] and S3VM [3]. SVM+ is a variant of SVM which addresses the issue of heterogeneous data. Specifically, SVM+ implicitly establishes a different hyper-plane for each data subgroup by modifying a standard SVM's objective function and constraints. S3VM is a semi-supervised version of

Yijun Zhao, Bryan Lackaye, Jennifer Dy and Carla Brodley "A Quantitative Machine Learning Approach to Master Students Admission for Professional Institutions" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 538 - 544

SVM which learns a classifier using both labeled and unlabeled data. Our contribution is a new variant of SVM that unifies the advantages of both S3VM and SVM+. Our new model, which we name S3VM+, addresses the admission biases in the labeled data and utilizes unlabeled applicants' data simultaneously. S3VM+ can be applied to any domain for which the data may have clearly defined subgroups (e.g., privileged domain knowledge) and a large amount of unlabeled data.

An additional motivation of our research was to validate our hypothesis of whether we could predict student success based only on quantitative measures and, thus, remove the subjectivity of the committee reading the recommendation letters and statement of purpose. If successful, such a model will not only lead to a better selected student body, but also help to manage growing enrollments. Our experimental results (see Section 4 for details) demonstrate that, with our new model, we can achieve an effective yet imperfect prediction. Thus, in practice, our model could serve as a Focus of Attention (FOA) tool for the admission committees.

The rest of the paper is organized as follows: in Section 2, we present the related work in predicting students' performance in the education domain. In Section 3, we give brief introductions to S3VM and SVM+ and present our model S3VM+ in detail. We demonstrate the efficacy of our model in Section 4 by comparing its performance to those three existing models. Finally, we conclude in Section 5.

2. RELATED WORK

Most EDM studies focus on predicting students' academic performance after they have been admitted to the college or program. For example, Lepp et al. investigated the relationship between cell phone use and academic performance in a sample of US college students [8]. Delen applied machine learning techniques for student retention management [6]. Ioanna et al. presented a dropout prediction in e-learning courses using machine learning techniques [10]. Nevertheless, another important aspect of educational research is selecting the best fitting students at admission time, which has not been widely addressed in past literature.

The most closely related work to our paper is the admissions research conducted by the University of Texas at Austin (UT Austin) for their graduate admission program [14], driven in part by their need to manage growing application numbers. In their work, the authors applied logistic regression (LR) to help the admission committee identify weak candidates who will likely be rejected and exceptionally strong candidates who will likely be admitted. Our work bears a similar mission but is different in three aspects. First, the UT Austin research includes credentials such as recommendation letters and statement of purpose, whereas our work strives to build a purely quantitative model relying only on non-subjective measures. Second, the recommendations made by UT Austin's algorithm are based on an applicant's likelihood of admission, whereas our model aims to predict the future performance of the applicants in the graduate program. Last, our model addresses human subjectivity in admission decisions. The contribution of our paper is a quantitative machine learning model to predict a candidate's future performance at admission time.

3. INTEGRATING SEMI-SUPERVISED SVM WITH DOMAIN KNOWLEDGE

We choose our model based on the characteristics of our dataset and particular challenges involved in our task. In particular, we choose SVM and two existing frameworks: S3VM [3] and SVM+ [12]), as our baseline models. Our proposed model is a new variant of SVM, which is inspired by S3VM and SVM+. We first give brief introductions to S3VM and SVM+. We then describe our new model in detail in Section 3.3.

3.1 S3VM (Semi-Supervised SVM)

S3VM is semi-supervised SVM proposed by [3]. The model is learned using a mixture of labeled data (the training set) and unlabeled data (the auxiliary set). The objective is to assign class labels to the auxiliary set such that the "best" support vector machine (SVM) is constructed. In particular, given a labeled dataset $L = \{x_1, x_2, \dots, x_l\}$ and an unlabeled auxiliary dataset $U = \{x_{l+1}, x_{l+2}, \dots, x_{l+k}\}$, S3VM learns a classifier from both L and U using overall risk minimization (ORM) posed by Vapnik [13] (Chapter 10). Starting with the standard SVM formulation, S3VM adds two constraints for each data point in the auxiliary set U . One constraint calculates the misclassification error as if the point were placed in class 1, and the other constraint calculates the misclassification error as if the point were placed in class -1. The objective function calculates the minimum of the two possible misclassification errors. The final membership assignments of the instances in U correspond to the ones that result in a minimum total sum of slacks across all instances in the training set. Specifically, we have:

$$\min_{w, b, \eta, \xi, z} \quad \frac{1}{2} \|w\|^2 + C \left[\sum_{i=1}^l \eta_i + \sum_{j=l+1}^{l+k} \min(\xi_j, z_j) \right] \quad (1)$$

subject to

$$\begin{aligned} y_i(w \cdot x_i + b) + \eta_i &\geq 1 & \eta_i &\geq 0 & i &= 1, \dots, l \\ w \cdot x_j + b + \xi_j &\geq 1 & \xi_j &\geq 0 & j &= l+1, \dots, l+k \\ -(w \cdot x_j + b) + z_j &\geq 1 & z_j &\geq 0 & j &= l+1, \dots, l+k \end{aligned}$$

where C is the trade-off between maximizing the margin and total violations. η_i 's are the slacks for the labeled data, and ξ_j 's and z_j 's are the slacks for the unlabeled data hypothetically assigned to the positive and negative classes respectively.

Equation (1) can be solved using mixed integer programming by applying the "large integer M " technique. The idea is to introduce a constant integer $M > 0$ and a decision variable $d_j \in \{0, 1\}$ for each point x_j in the auxiliary set U . d_j indicates the class membership of x_j . If $d_j = 1$, then the point is in class 1 and if $d_j = 0$, then the point is in class -1. The integer M is chosen sufficiently large such that if $d_j = 0$ then $\xi_j = 0$ is feasible for any optimal w and b . Likewise if $d_j = 1$, then $z_j = 0$. In other words, ξ_j and z_j can have at most one non-zero value no matter what class x_i belongs to. Consequently, we could replace the $\min(\xi_j, z_j)$ in Equation (1) by

$(\xi_j + z_j)$. This results in the following formulation:

$$\min_{w, b, \eta, \xi, z} \quad \frac{1}{2} \|w\|^2 + C \left[\sum_{i=1}^l \eta_i + \sum_{j=l+1}^{l+k} (\xi_j + z_j) \right] \quad (2)$$

subject to

$$\begin{aligned} y_i(w \cdot x_i + b) + \eta_i &\geq 1 \\ \eta_i &\geq 0, \quad i = 1, \dots, l \\ w \cdot x_j + b + \xi_j + M(1 - d_j) &\geq 1 \\ -(w \cdot x_j + b) + z_j + Md_j &\geq 1 \\ \xi_j &\geq 0, \quad z_j \geq 0, \\ j &= l+1, \dots, l+k, \quad d_j \in \{0, 1\} \end{aligned}$$

The solution to Equation (2) can be found using mixed integer programming products. In our experiment, we used CVX [1] and Gurobi [2] optimizers. Same as a standard SVM, S3VM classifies a new instance x^* using $\text{sign}(w \cdot x + b)$.

3.2 SVM+

Vapnik and Vashist [12] introduced SVM+, which is a variant of SVM that addresses the issue of learning with heterogeneous data. In their model, the authors developed a new paradigm to learn using privileged information (LUPI). The objective of SVM+ is to take advantage of additional domain knowledge, and in particular data subgroups that may arise from different sources or due to labeling biases.

Suppose the training data has $t > 1$ groups. We follow the notation in [9] and denote the indices of group r by

$$T_r = \{i_{n_1}, \dots, i_{n_r}\}, \quad r = 1, \dots, t$$

All training samples can then be represented as:

$$\{\{X_r, Y_r\}, \quad r = 1, \dots, t\}$$

where $\{X_r, Y_r\} = \{(x_{r_1}, y_{r_1}), \dots, (x_{r_{n_r}}, y_{r_{n_r}})\}$. To incorporate the group information, SVM+ defines the slacks inside each group by a unique *correcting function*:

$$\xi_i = \xi_r(x_i) = \phi_r(x_i, w_r), \quad i \in T_r, \quad r = 1, \dots, t$$

Specifically, the correcting functions are defined as:

$$\xi_r(x_i) = w_r \cdot x_i + d_r, \quad i \in T_r, \quad r = 1, \dots, t$$

Compared to a standard SVM, S3VM uses slack variables that are restricted by the correcting functions, and the correcting functions capture additional information about the data. Note that all of the data is used to construct the decision hyperplane. The group information is only used to fine tune the slack variables. Formally, the objective function for SVM+ is formulated as follows:

$$\min_{w, b, w_1, w_2, w_r, d_1, d_2, d_r} \quad \frac{1}{2} \|w\|^2 + \frac{\gamma}{2} \sum_{r=1}^t \|w_r\|^2 + C \sum_{r=1}^t \sum_{i \in T_r} \xi_i^r \quad (3)$$

subject to:

$$\begin{aligned} y_i(w \cdot x_i + b) + \xi_i^r &\geq 1 \\ \xi_i^r(x_i) &= w_r \cdot x_i + d_r \\ \xi_i^r &\geq 0, \quad i \in T_r, \quad r = 1, \dots, t \end{aligned}$$

Parameter γ adjusts the relative weight between $\|w\|^2$ and the $\|w_r\|^2$'s. C is the trade-off between maximizing the margin and total violations.

Liang and Cherkassky [9] further extended the SVM+ approach to multi-task learning. In the SVM+MTL [9] framework, the data is partitioned into groups using privileged information similar to the SVM+ model. However, instead of making a correcting function for the slack variables, their model establishes a unique correcting function (i.e., a hyper-plane) for each group in addition to a shared common hyper-plane. In other words, the decision function for group $r = 1, \dots, t$ is as follows:

$$f_r(x) = (w \cdot x + b) + (w_r \cdot x + d_r)$$

where w, b are the parameters for the common hyper-plane and w_r, d_r are the parameters for the correcting function for group r . The corresponding formulation of the quadratic optimization problem is as follows:

$$\min_{w, b, w_1, w_2, w_r, d_1, d_2, d_r} \quad \frac{1}{2} \|w\|^2 + \frac{\gamma}{2} \sum_{r=1}^t \|w_r\|^2 + C \sum_{r=1}^t \sum_{i \in T_r} \xi_i^r \quad (4)$$

subject to

$$\begin{aligned} y_i[(w \cdot x_i + b) + (w_r \cdot x_i + d_r)] + \xi_i^r &\geq 1 \\ \xi_i^r &\geq 0 \quad i \in T_r, \quad r = 1, \dots, t \end{aligned}$$

SVM+MTL is an adaptation of SVM+ for solving MTL problems. In our experiment, we applied the SVM+MTL framework because it provides more flexibility to learn a different decision plane for each year's student data.

For SVM+MTL, predicting the class label for a new given instance x^* is not straightforward because its decision function requires a group-dependent correcting function, and we do not know the group membership of test instances. To resolve this problem, we predict the label for x^* in each group and perform a majority vote over all predicted labels. Specifically, a test instance x^* will be predicted in each group as follows:

$$f_r(x^*) = \text{sign}[(w \cdot x^* + b) + (w_r \cdot x^* + d_r)]$$

where $r = 1, \dots, t$ are the bias groups, and w, b, w_r 's and d_r 's are learned model parameters. The class membership for x^* is determined by a majority vote over $f_r(x^*)$'s.

3.3 S3VM+

Our new model, S3VM+ leverages the unlabeled data and addresses the biases in the training data simultaneously. In particular, we train our model with a labeled dataset and an unlabeled auxiliary dataset. Furthermore, our data is partitioned into yearly groups because of the admissions committee changes from year to year and thus may have different biases. For the labeled dataset, we incorporate the grouping information by establishing a correcting function for each group (constraints (a) and (b) in Equation (5)).

For the unlabeled data, we introduce two slack variables ξ_i and z_i for each data point x_i representing the slacks of placing x_i in the positive class and negative classes respectively. The objective function for S3VM+ takes the minimum of the two slacks for each unlabeled instance and minimizes the total sum of slacks across all training instances. We apply the "large integer M " technique (see Section 3.1 for details) and convert the constraint with a minimization function to two constraints over linear functions. Because both labeled and

unlabeled data are grouped by academic year, we apply the same correcting functions used for the labeled data to each corresponding annual group of unlabeled data (constraints (c) to (f) in Equation (5)). Formally, the optimization problem for S3VM+ is formulated as follows:

$$\min_{w, b, w_1, w_2, w_r, d_1, d_2, d_r} \frac{1}{2} \|w\|^2 + \frac{\gamma}{2} \sum_{r=1}^t \|w_r\|^2 + C \left[\sum_{i=1}^l \eta_i^r + \sum_{j=l+1}^{l+k} (\xi_j^r + z_j^r) \right] \quad (5)$$

subject to

- (a) $y_i[(w \cdot x_i + b) + (w_r \cdot x_i + d_r)] + \eta_i^r \geq 1$
- (b) $\eta_i^r \geq 0 \quad i = 1, \dots, l$
- (c) $[w \cdot x_j + b + (w_r \cdot x_j + d_r)] + \xi_j^r + M(1 - d_j) \geq 1$
- (d) $\xi_j^r \geq 0 \quad j = l+1, \dots, l+k, \quad d_j \in \{0, 1\}$
- (e) $-[(w \cdot x_j + b) + (w_r \cdot x_j + d_r)] + z_j^r + Md_j \geq 1$
- (f) $z_j \geq 0 \quad j = l+1, \dots, l+k \quad d_j \in \{0, 1\}$

where C is the trade-off between maximizing the margin and total violations, and γ is the trade-off parameter between $\|w\|^2$ and the $\|w_r\|^2$'s. Note that constraints (a), (b) are for labeled instances and constraints (c) – (f) are for unlabeled instances.

To classify a new instance x^* , we follow the same approach as SVM+, which is to take a majority vote on class labels predicted by each group.

4. EXPERIMENTAL RESULTS

In this section, we first describe the process of constructing our training and testing dataset. We then discuss the methods we used to conduct our experiments in Section 4.2. We present our analysis of our experiments in Section 4.3.

4.1 Constructing the Training and Test Data

We have a real world dataset consisting of students from the MSCS program at Northeastern University. Table 1 presents the features we collected from students' applications for our experiment. Feature 1 contains the students' undergraduate GPAs adjusted according to each individual university's grading scale. For example, a 3.5 out of 5 and a 7 out of 10 would result in the same value. Feature 10 contains self-reported values representing the maximum number of lines of programming written by the student prior to joining the MS program. Feature 12 contains the rankings of the undergraduate institutions where the students obtained their bachelor's degrees. We classified the rankings into 4 categories with 1 being the most prestigious. The classification was performed manually according to the Best Global Universities list published by US News and World Report. The rest of the features are standardized test scores. Both the GRE and TOEFL had two versions of tests during 2009 - 2012 which use different scoring scales. Both of these tests are converted to their new versions of scoring scales using conversion tables provided by the ETS [4].

As mentioned in Section 1, our task is to identify successful candidates at the point of admission. One measure of success is MS-GPA in the MS program (as distinct from the input feature 1 "Undergraduate GPA"). Indeed, a cumulative MS-GPA is the most widely used measure for students'

Table 1: Features Collected for Training

1	Undergraduate GPA
2	GRE Verbal
3	GRE Quantitative
4	GRE Analytical Writing
5	TOEFL Total
6	TOEFL Reading
7	TOEFL Listening
8	TOEFL Speaking
9	TOEFL Writing
10	Max # of Lines of Code Written
11	Bachelor's Degree in EECS (Yes/No)
12	Undergraduate School Ranking

Table 2: Student Data Statistics

Year	Total	Top 20%	Bottom 20%	Aux. Data
2009	37	7	7	431
2010	89	18	17	503
2011	132	28	27	705
2012	196	51	42	948

academic performance [11]. The labels in our training data are determined by the training instances' percentiles in the overall MS-GPAs. Specifically, the top and bottom 20% students are labeled with class 1 and -1 respectively. The number 20% was intuitively chosen as an measure which sets the individuals apart from the average students.

Note that we did not use a midpoint MS-GPA as a cutoff to separate the positive and negative classes, in order to reduce the label noise. In particular, instances close to the average GPA are harder to categorize as good or bad students.¹ Another intuitive approach is to define two hard MS-GPA thresholds for good versus bad performances, i.e., to have a MS-GPA above an upper threshold (e.g., > 3.8) for good students, and below a bottom threshold (e.g., < 3) for bad students. A further investigation reveals that this approach is less effective for the following reason: different instructors have different grading policies due to the nature of the courses. For some fundamental courses, an 'A' means you are in the top 30% of a class, while for some other advanced courses, an 'A' means you are in the top 10% of a class. Even for the same course in the same year with different sections, the instructors may choose to cooperate exams/grading or not. Because students have different instructors and/or even take different courses, hard cutoffs are not an accurate reflection of a student's abilities.

Having stated this, on the other hand, if a student performs consistently in the top 20% in each class, this student will be among the top 20th percentile of the entire MS-GPA spectrum. The same can be said for those that perform consistently in the bottom 20th percentile. Identifying the factors that lead to this consistent success or underperformance are of greatest interest to this research. Therefore, we used relative measures to label our positive and negative training samples. For comparison purposes, we report our experimental results on both relative and hard cutoffs in Tables 5 and 6 respectively.

¹We did experiment with splitting the two classes using the mean value of all MS-GPAs and the performance was not satisfactory as expected.

Table 3: Prediction Using 1Y Data

Train	Test	Top20% MS-GPA	Bot20% MS-GPA	Overall
2009	2010	0.72	0.59	0.66
2010	2011	0.64	0.70	0.67
2011	2012	0.65	0.76	0.70

Table 4: Predicting Using 10-fold Cross Validation

Model \ MS-GPA%	Test Accuracy			Training Accuracy		
	Top20	Bot20	Overall	Top20	Bot20	Overall
2009 - 2011	0.70	0.71	0.71	0.79	0.79	0.79
2009 - 2012	0.74	0.72	0.73	0.84	0.75	0.79

Table 2 summarizes the distribution of students from 2009 to 2012. Column “Total” is the total number of students enrolled in the corresponding year. Columns “Top 20% MS-GPA” and “Bottom 20% MS-GPA” are the total number of students in the top and bottom 20th percentile among their peers measured by the cumulative MS-GPAs. There is not an equal number of positive and negative instances for each year because there are multiple students with same MS-GPA.

Both SVM+ and our model S3VM+ make use of an unlabeled auxiliary dataset. We collect the application data of rejected (i.e., not admitted) and declined (i.e., admitted but not enrolled) applicants as the auxiliary data. These data contain the same features as the labeled data, and the size distribution of auxiliary data from 2009 to 2012 is presented in the last column of Table 2. Our training data are all labeled and unlabeled instances from 2009 to 2011, and our test data are labeled instances from 2012.

4.2 Experimental Method

We are interested in identifying the top and bottom 20% of candidates from an application pool based on the performance of the admitted students. Our first goal is to confirm our conjecture that there are biases in admission decisions from year to year. To this end, we conducted two experiments. The first experiment is to use the previous year’s data to predict the current year’s performance using a standard SVM. For example, we would use class 2009’s data to predict class 2010’s performance, and class 2010’s data to predict class 2011’s performance. Table 3 presents the prediction accuracies for each year. We observe that, for 2010, the top 20% of students are easier to predict than the bottom 20%, whereas for 2011 to 2012, the situation is reversed. This lack of consistency and the low overall accuracies (up to 70%) suggest that there is no strong correlation of predictive patterns from year to year. Our second experiment is to apply a standard 10-fold cross validation on two datasets: all data from 2009 to 2011 and all data from 2009 to 2012. Because 2012 added a significant amount (89%) of instances, we would expect a noticeable increase in both the training and test accuracies if the data across different years conform to the same distribution. Table 4 summarizes the results of this experiment. We observe only a marginal improvement in overall test accuracy after adding instances from 2012 and, more importantly, the overall fit of the data remains the same (79%). From these two experiments, we conclude

Table 5: Performance Comparison with Relative Cutoffs

Model \ MS-GPA%	Test Accuracy			Training Accuracy		
	Top20	Bot20	Overall	Top20	Bot20	Overall
SVM	0.73	0.71	0.72	0.79	0.80	0.79
S3VM	0.75	0.74	0.74	0.81	0.82	0.81
SVM+	0.77	0.70	0.74	0.92	0.84	0.88
S3VM+	0.82	0.72	0.77	0.95	0.89	0.92

Table 6: Performance Comparison with Hard Cutoffs

Model \ MS-GPA	Test Accuracy			Training Accuracy		
	>3.8	<3.4	Overall	>3.8	<3.4	Overall
SVM	0.65	0.69	0.66	0.73	0.75	0.74
S3VM	0.72	0.65	0.70	0.83	0.70	0.77
SVM+	0.75	0.64	0.71	0.92	0.75	0.84
S3VM+	0.77	0.67	0.74	0.93	0.80	0.87

that data across different academic years have different distributions. We believe this year to year bias is due to the change in the membership of the admission committee.

In light of above learned information, we partitioned the data by academic year and use them as the privileged groups in SVM+ and S3VM+. We take the union of labeled data from 2009 to 2011 as our labeled training data. The auxiliary dataset is formed as the union of the corresponding auxiliary data from 2009 to 2011. We test and compare the performance of the four models (SVM, S3VM, SVM+, S3VM+) in predicting labeled instances in 2012.

The hyper-parameters are the trade-off constant C for all four models and γ for SVM+ and S3VM+. We perform 10-fold cross validation and grid search on the training data to select the hyper-parameters. We first use a coarse grid $\{0.01, 10, 1000\}$ for C and refine the candidates after the initial search. The final list for C is $\{1, 10, 100\}$. Following a similar procedure, our final search list for γ is $\{0.01, 1, 100\}$. After the best hyper-parameters are selected, we train the corresponding model one more time using the entire training data and then apply the learned model to the test data and measure its performance. We report both training and test accuracies in Table 5.

4.3 Analysis on Performance Measures

Table 5 displays the main results of our experiment. First, we observe that the test accuracies for SVM on the positive and negative classes are more balanced compared to the results in Table 3. There is also an improvement in the overall performance for SVM. This can be explained by the increased amount of training data used in our Table 5 experiment.

Second, we conclude that all three variants of SVM (S3VM, SVM+, S3VM+) are superior to standard SVM. Using SVM as a baseline measure:

- S3VM improved slightly on the accuracies of both positive and negative classes, which suggests that using auxiliary data has a positive impact on identifying both the good and bad students. This is consistent with the fact that the auxiliary data contain both de-

clined (i.e., admitted but not enrolled) and rejected (i.e., not admitted) applicants, which could improve the accuracy of positive and negative classes respectively.

- SVM+ demonstrated improvement on the positive side only, which indicates that the partition of bias groups by academic year is most effective in identifying the top students. One explanation for this could be that the top 20% of students are inherently different from year to year, while the bottom 20% of students remain similar. Or that a particular admissions committee has biases about how to recognize a strong student.
- Our model S3VM+ has a noticeable advantage among all models in predicting the positive class: 83% versus 73% (SVM), 75% (S3VM) and 77% (SVM+). In light of the construction of S3VM+, one could conclude that adding auxiliary data to each partition group further enhances the power of identifying top students. On the other hand, because grouping does not have a significant impact on identifying bottom students (as demonstrated by SVM+), S3VM+ would only result in a limited gain for the negative class.

Lastly, from the training accuracies presented in Table 5, we observe a significantly better fit of the training data using our model S3VM+. In particular, 95% versus 92% (SVM+), 81% (S3VM), 79% (SVM) accuracies for the positive class and 89% versus 84% (SVM+), 82% (S3VM) and 80% (SVM) accuracies for the negative class. Compared to the standard SVM, S3VM improved training accuracies evenly on both classes, and SVM+ and S3VM+ demonstrated more significant gains on the positive class, which is consistent with what we observed in the test data.

4.4 Labeling Strategy: Relative v.s. Absolute

Recall that in Section 4.1, we discussed our choice of labeling the top 20% and bottom 20% of students with respect to their MS-GPAs as our two classes. We explained our rationale of using relative rather than hard cutoffs to label our data. We confirm this conjecture in Table 6, where we show the results of an experiment using $\text{MS-GPA} > 3.8$ for the top students and $\text{MS-GPA} < 3.4$ for the other. In the table we see that for all four methods, the overall accuracies are lower than in Table 5.

4.5 Analysis on Weight Vectors

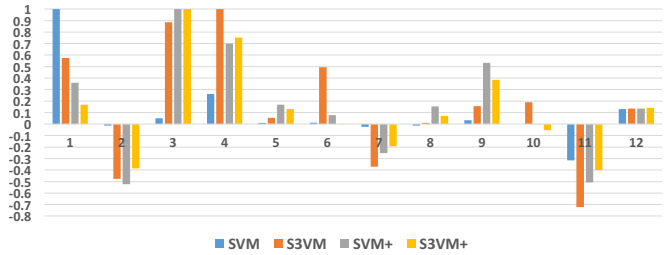
Because we utilized a linear SVM and its variants, we found it interesting to investigate the ranking and magnitude of each individual feature in the weight vectors produced by each model. Table 7 presents the ranking of w_i 's in the weight vectors (w 's) of four models. Figure 1 displays the weights of individual features across four models using their magnitudes. In order to make a meaningful comparison, each weight vector $w = \{w_1, w_2, \dots, w_{12}\}$ is scaled by the maximum absolute value of its components. Thus, the weight for the most important feature is either 1 or -1. Note that, for SVM+ and S3VM+, we display the shared hyper-plane vector w without the correcting functions for each group.

From Table 7, we observe that all models except standard SVM suggest the same top two features: "GRE Quantitative" and "GRE Analytic Writing" scores. Furthermore,

Table 7: Weights Ranking Comparison

Model	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
SVM	w_1	w_{11}	w_4	w_{12}	w_3	w_9	w_7	w_2	w_6	w_8	w_5	w_{10}
S3VM	w_4	w_3	w_{11}	w_1	w_6	w_2	w_7	w_{10}	w_9	w_{12}	w_5	w_8
SVM+	w_3	w_4	w_9	w_2	w_{11}	w_1	w_7	w_5	w_8	w_{12}	w_6	w_{10}
S3VM+	w_3	w_4	w_{11}	w_2	w_9	w_7	w_1	w_{12}	w_5	w_8	w_{10}	w_6

Figure 1: Weights Distribution Over 12 Features Across Four Models



The weights are normalized using $w = \frac{w}{\max_{1 \leq i \leq 12} \{|w_i|\}}$

SVM+ and S3VM+ overlapped in their top five features but with a different ranking order.

From Figure 1, we conclude that the most important features are 1 ("Undergraduate GPA"), 2 ("GRE Verbal"), 3 ("GRE Quantitative"), 4 ("GRE Analytical Writing") and 11 ("Bachelor's Degree in EECS (Yes/No)"). A closer examination reveals that SVM relies mostly on three features (1, 4, and 11). S3VM has significantly large weights on two additional features, 6 ("TOEFL Reading") and 7 ("TOEFL Listening"), on top of the five features listed above. SVM+ and S3VM+ made use of one additional feature which is 9 ("TOEFL Writing").

5. CONCLUSIONS

In this paper, we applied a quantitative machine learning approach to predict candidates' potential academic performances based on information from their applications. We built our model using empirically admitted students with their cumulative GPAs as performance measures and tested our model's efficacy for the incoming students. Throughout our experiments, we found a unique challenge associated with our task, which is different data distributions across the academic years due to biases arising from changing membership of the admissions committee. We addressed this issue with the Learning Using Privileged Information (LUPI) framework. We further handled the limited training data issue by employing a semi-supervised version of SVM to utilize the large amount of unlabeled data (i.e., the rejected/declined applications). Our resulting model, S3VM+, is a novel variant of SVM that addresses subjectivity and lack of labeled data simultaneously. Our experimental results demonstrate a significant gain of our model compared to three existing models in standard literature (i.e., standard SVM, S3VM, and SVM+). Although we based our work on a two-year master's program, our model is easily extensible to similar tasks such as college or pre-school admissions. Our model can also be applied to other real world situations in which data may have clearly defined biased subgroups and a large amount of unlabeled data.

6. REFERENCES

- [1] *CVX Research Inc.* www.cvxr.com.
- [2] *Gurobi Optimizer.* www.gurobi.com.
- [3] K. Bennett and A. Demiriz. Semi-supervised support vector machines. *NIPS*, 11:368–374, 1998.
- [4] CGS and ETS. Graduate enrollment and degrees: 2005 to 2015. <http://cgsnet.org/reports>, 2016.
- [5] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20.3:273–297, 1995.
- [6] D. Delen. A comparative analysis of machine learning techniques for student retention management. *Decision Support Systems*, 49.4:498–506, 2010.
- [7] EDM. International educational data mining society. <http://www.educationaldatamining.org/>, 2016.
- [8] A. Lepp, J. E. Barkley, and A. C. Karpinski. The relationship between cell phone use and academic performance in a sample of us college students. *Sage Open*, 5.1:2158244015573169, 2015.
- [9] L. Liang and V. Cherkassky. Connection between svm+ and multi-task learning. *IEEE World Congress on Computational Intelligence*, pages 2048–2054, 2008.
- [10] I. G. V. N. G. M. Lykourantzou, Ioanna and V. Loumos. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Computers and Education*, 53, no. 3:950–965, 2009.
- [11] A. M. Shahiri and W. Husain. A review on predicting student’s performance using data mining techniques. *Procedia Computer Science*, 72:414–422, 2015.
- [12] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22.5:544–557, 2009.
- [13] V. N. Vapnik. *Estimation of dependences based on empirical data*. New York: Springer-Verlag, 1982.
- [14] A. Waters and R. Miikkulainen. Grade: Machine learning support for graduate admissions. *AI Magazine*, 35.1:64, 2014.

Where to aim? Factors that influence the performance of Brazilian secondary schools

Paulo J.L. Adeodato
Universidade Federal de Pernambuco
Centro de Informática
Recife - Brazil
pjl@cin.ufpe.br

Rogério L. C. Silva Filho
Universidade Federal de Pernambuco
Centro de Informática
Recife - Brazil
rlcsf@cin.ufpe.br

ABSTRACT

There have been discussions on where to invest the budget allocated to education. Most politicians want to invest in the schools' infrastructure, but is that the most efficient policy for spending? This paper presents analyses to help clarify that. It integrates the most recent data sources (2018) on the secondary students' assessment (ENEM), the School Census and the Teachers' Census and consolidates all microdata to the school level, making them features of the schools. These features are then grouped into three types: infrastructure, human education and socio-economic aspects. Then the features from each group are applied in logistic regression predictive models both isolate and collectively. In a 10-fold cross-validation comparison with the area under the ROC curve as metric. The experimental results show that infrastructure is significantly less influential than the other features. Further research needs to consider investment costs and time to produce effect on school performance.

Keywords

School quality assessment, Educational decision support system, Educational Data Mining, Domain-Driven Data Mining, Educational budget allocation

1. INTRODUCTION

International comparison of students' performance among countries by the Programme for International Student Assessment (PISA) has yielded strategic discussions in international education policies. The PISA, sponsored by Economic Co-operation and Development (OCDE), aims at assessing and providing a global perspective on secondary education (15-year-old pupils) across countries of the world [16].

Following the international efforts, the local governments have been concerned with standardized tests themselves, aiming at the assessment of students as much as at monitoring the quality of the educational system [1]. In Brazil, the National Institute for Educational Studies (Instituto Na-

cional de Estudos e Pesquisas Educacionais – INEP) produces the annual School Census which is a survey of the schools for secondary education in the country and the National Secondary School Exam (Exame Nacional do Ensino Médio – ENEM) that evaluates student performance at end of secondary education.

In 2009, ENEM became a mechanism for students' admission to higher education in public universities. That improved the quality of the information collected. Added to the technical knowledge of each student, ENEM also captures their socio-economic-cultural (SEC) information [2]. The integration of this information with the School Census data has become a relevant source of data for scientific studies and enables the Federal Government to define and validate public policies for Brazilian education [30]. However, secondary education is under jurisdiction of the constituent states of the federation, not the national government. Thus, despite the importance of the federal government role, there is considerable variation among the states in curriculum, teacher training, budget policies and other issues [10].

Many factors can influence the performance of the students. Studies have shown that school inputs, students' SEC background, parents' education are correlated with student achievement [13, 7, 12]. In Brazil, according to the last school census available for this research (2018), 42% of the secondary public schools still lack Basic Infrastructure Level. The definition of the quality of the levels was performed by Neto [26] being the Basic Level the second lowest of four levels which includes features like having a management room with computer and printer for administrative work only. This scenario makes Brazilian politicians focus most of their educational bills and budget allocations on improving school infrastructure [19].

Despite the importance of providing infrastructure to schools, it is common for politicians to invest in infrastructure such as computer labs, tablets, TVs etc. even for schools that have not reached the basic level yet. This paper does not discuss pedagogical issues related to infrastructure; just tries to help policymakers and education-related institutions on how to invest their budget to comply with both regulations and education quality goals in a long-run plan to secondary public education.

This paper presents experiments in a Domain-Driven Data Mining (DM) approach that assesses the quality of secondary

Paulo Jorge L. Adeodato and Rogerio Luiz C. S. Filho "Where to aim? Factors that influence the performance of Brazilian secondary schools" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 545 - 549

schools in Brazil. The results show that the infrastructure predictors are less relevant than SEC and educational predictors at a 5% significance level. Experiments were carried out on the most recent data available (2018) with logistic regression models in a 10-fold cross-validation setting.

The paper is organized in 4 more sections. Section 2 presents the data sources and preprocessing. Section 3 describes the experimental project to elucidate the most influential group of features for predicting good students. Section 4 presents the results and discusses its impacts. And Section 5 presents the conclusions, difficulties found and suggestions for future work.

2. DATA SOURCES AND PREPROCESSING

This research has used two official public databases: Microdata from the National Secondary School Exam 2017 and 2018 containing the students SEC information and their grades on the test at the end of secondary education, and the School Census 2018 [2] detailing the conditions of the schools, from physical infrastructure to faculty information. The 2017 ENEM database was only used as an independent statistical sample to apply the process of granularity transformation described in Subsection 2.4. These databases refer to over 5 millions of students of 32,000 secondary schools across the country, but this paper will focus only on public (free) schools.

2.1 The Universe of Schools (Scope)

This research attempts to help policymakers optimize the budget allocation in order to improve Brazilian Secondary Schools. That does override the priority of the 42% of public secondary that have only an elementary infrastructure (just classrooms, electric energy, sanitation and piped water). A few schools (0.7%) were discarded from the database for being below that level.

ENEM is a democratic exam that any person can sit. That makes it necessary to apply some selection filters in student grain: a) Students who have no school assigned, are just training or are not in the last secondary school year (74%), b) students who do not follow a regular curriculum (2.5%) and c) foreign students (0.02%). The remaining 680,583 students were considered. To eliminate anomalies that could either divert from the goal or deteriorate the quality of the work, students who did not perform all the tests, including the essay, were also left out of the scope of this research.

Back to the school grain, for having critical mass, only schools with 10 or more students were selected, as established by INEP in the analyses. After this last filter, the total that remained in this research dropped to 14,579 secondary schools with 653,848 students which form the dataset used in this paper's experiments.

2.2 Problem Characterization and Goal Setting

In business, one of the most common decision strategies for selecting the eligible candidates for an action is ranking them according to a classification score and choosing those above a predefined threshold [17]. That is used in applications such as staff selection, fraud detection [6], and resources allocation

in public policies, for instance. This score is computed by either weighing a set of variables based on human-defined parameters or by applying a function learned by a classification algorithm from a set of data with binary labels as desired response, according to specific optimization criteria.

In some domains of application, several problems are ill-defined simply because stakeholders do not reach consensus on either method [29]. That is particularly true for education where experts and faculty do not agree even on the characterization of a good school or a good student. To circumvent these issues, we have adopted the systematic approach proposed by [3] to characterize this as a binary decision problem. Thus, the problem can be solved by machine learning algorithms based on the supervised learning paradigm with a data dependent strategy where each example is labeled as "good" or "bad" for binary decision making. That involves solving two scientific issues which represent controversial points in the application domain: (1) which metrics should be used as a ranking score for evaluating the quality of the school and (2) which threshold should be adopted as a criterion to define what would be a "good" school in the binary decision.

The ENEM [1] has been conceived to assess the quality of the Brazilian secondary schools based on their students' evaluation on the test. Despite arguments among experts on education, they have agreed that the performance of the students at the last year would represent their performance in the secondary school and also agreed that the mean student score would be the most relevant indicator of each school, as already done in previous studies [3, 30].

2.3 Binary Goal Definition

Once having defined the quality metrics, the most controversial point is to set the threshold to characterize what would be a "good" or "bad" school in the dichotomic objective. Once again, to circumvent the controversy and lack of consensus in the field on the issue and bring a higher level of abstraction that enables future comparison across years, regardless of the degree of difficulty of the exams, this study used statistics concepts for setting the threshold as recommended by [28]. Quartiles of the distributions not only are robust against extreme values (outliers) [21], but also can be a straightforward data dependent dichotomizing criterion of interest for the application domain. The upper quartile has already been successfully used as threshold [28] on a continuous goal variable for creating a binary target-variable. This paper has adopted that approach for converting the problem into a binary classification where the upper quartile represents the "good" schools.

2.4 Granularity Transformation

The granularity of the attributes is a fundamental concept and its diversity brings great complexity to research of this nature. How can one associate to each school its family income attribute from the distribution of family income of their students? How can one associate to each school its faculty education attribute from the distribution of faculty education of their teachers? These transformations represent a difficulty for teams without professionals specialized in developing data mining projects. This difficulty is due to both the sheer volume of data to be handled and the

need to use artificial intelligence to embed knowledge of experts in education in the transformation of the attributes for granularity change in a process coined Domain-Driven Data Mining (D3M)[23].

We considered and chose the Regression Granularity Transform (RGT) [4] as the most adequate approach for this research. It aims at maximizing the information gain towards the target class for categorical microdata present in Student and Teacher grains. Logistic Regression was the technique applied on the categorical attribute distribution having its histogram with the categories' relative frequencies as input. These transformations were learned from the previous year data (2017), to avoid having to discard data from the focus year of 2018. For numerical features, the average was the transformation adopted.

2.5 Preprocessing

Many factors affect the success of a data mining application. Data quality is among them [20]. Domain and data understanding allowed for the removal of irrelevant attributes (e.g. linked to elementary and fundamental schools and to other secondary school models that do not use the regular curriculum), attributes with a posteriori information and identification codes.

In the final data sample, just two binary features presented missing values, which were filled with "0", because they represented lack of that property. The categorical features that had the mode representing over 90% of the cases were removed.

For features with correlation higher than 0.8, only those with the highest semantic value for the domain were preserved. To reduce the influence of outliers and improve the quality of the Logistic Regression models, all numerical features were normalized using the α -winsorized values of the distribution ($\alpha/2 = 0.025$ at each tail) as their minimum and maximum.

3. EXPERIMENTAL PROJECT

The experiments were carried out using the Logistic Regression model in a 10-fold cross-validation setting. The features on school grain were partitioned into 3 different groups: 1) Infrastructure of schools, 2) SEC information of Students and 3) Level of Education of Parents and Teachers. The same held-out fold was used as test dataset for all groups and the models' performance on it was assessed by Area Under the Receiver Operating Characteristic (ROC) curve. ROC curve plots the true positive rate against the false positive rate, at all possible decision thresholds.

The goal is to experimentally compare the discriminant power of each group of predictors, focusing on groups 1 and 3, once that it is hard to produce any change in group 2 with educational policies. In one hand, it is widely known that features from group 3 are more influential than those of group 1, but it is hard for education policymakers to intervene on that due to limitations on either the country's economy or the Cash Transfer policies [22]. On the other hand, investment in features from group 1 has been the main focus of government, either by the insufficient conditions in some schools or because these investments have their effect more easily assessed. There are some studies in the literature on public

policies addressing teachers training and parents education [5, 8]. This paper aims at showing that the predictors of group 3 are more influential in predicting performance than those of group 1.

3.1 Performance analysis

Figure 1 shows the results for each test set in the 10-fold cross-validation process. In turns, one partition (fold) is separated for testing while the other 9 are used for training the model. The performance of the ROC curve at each fold, the average and the standard deviation across the 10 folds are the values reported. By comparing the results of groups 1 and 3, in the one-sided paired t-test, we accept the alternative hypothesis that the mean of the education group (3) is greater than that of the infrastructure group (1) at 0.05 significance level.

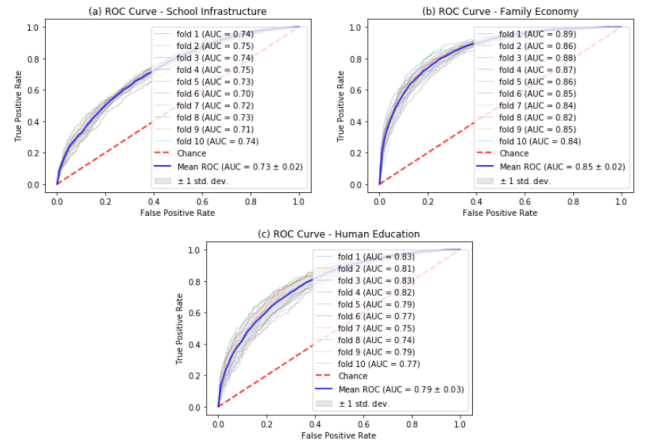


Figure 1: represents (a) the outputs for schools' infrastructure group, (b) the outputs for students' socioeconomic-cultural information group and (c) parents and teachers' level of education group.

The difference is highlighted even more when analyzing the number of variables in each group. Group 1 has 23 variables that represent the school structure while group 3 has only 3 variables, namely, the level of education of fathers, mothers and teachers. Analyzing group 3 in a logistic regression model on the whole dataset for assessing the features' influences according to their β coefficients, their predictive powers were in decreasing order, the father's education, the mother's education and the teachers' education. The qualification of teachers, in contrast to existing studies [18], does not have high explanatory power. This result is probably due to the fact that, in Brazil, the number teachers with M.Sc. and Ph.D. degrees in public secondary education is minimal (4.8% and 1.1%, respectively). Table 1 displays the beta coefficients of each variable and their p-value, well below the 0.05 significance level.

4. DISCUSSION

Several studies have improved the understanding of the determinants of school performance with the perspective of guiding educational policies. James Coleman, in 1960, had already identified the SEC factors of the students as the main determinant of their performance [13]. The correlation between parental education has been long established,

Table 1: Variable importance of the Logistic Regression model for group 3

Feature	β	p-value	Grain
Father's Education	3.87	0.00	Student Level
Mother's Education	2.94	0.00	Student Level
Teacher's Education	2.62	0.00	Teacher Level

as well [15, 11]. Other lines of research have also highlighted the relevance of aspects related to schools and teachers [18, 25, 14]. Much has been discussed in Brazil about the secondary education, as well as improving the budget allocation.

According to OECD, Brazil's public spending on education was close to the average of its member countries in the year of 2015 while the performance of Brazil in the last PISA exam was among the worst countries evaluated [27]. The quality of education still does not respond to the investments made. Therefore, it is crucial to improve the understanding of standardized national tests to help policymakers and education related institutions in developing educational public policies to produce an effective return on investment.

4.1 Parents Education as Proxy to SEC?

Separating out the independent effects of family education and SEC background is not a simple task. Some prior studies showed that those features are very correlated to family income, once parents who are more educated, earn higher salaries [24]. From another perspective, more education empowers parents and teachers to give the students better counseling and training. Some studies have tried to isolate the effect of each feature, aiming at determining causal relations between them in the educational outcomes [9].

This Subsection attempts to dissociate these characteristics to find out if the parents' education influences the student performance in the ENEM Exam for families with the same constant income. We started by considering only students from schools with infrastructure at basic level or above. To block any effect of economics, the students were undistinguishable by their family income which was kept constant. The performance was measured by the fraction (percentage) of good students in the sample, for each level of education.

Figure 2 shows the fraction of good students against the parents' education for each income value. It is clear that higher parents' education is associated with higher fraction of good students, with the income constant. Despite being a categorical feature, the level of education is associated with time of schooling, therefore suited to line graph representation.

Wrapping up, the students' performance on ENEM increases with their parents' level of education no matter their family income.

5. CONCLUSION

This paper has presented a comparative study of the influence of groups of predictors in the quality assessment of secondary school in Brazil to help policy makers in educational budget allocation.

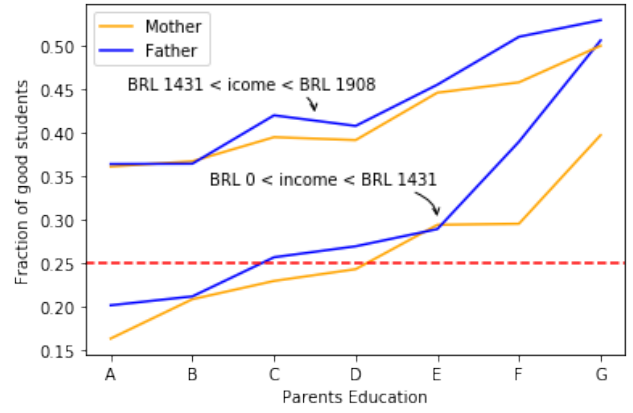


Figure 2: Fraction of good students as function of their parents' education level. Ranges from "A" (no schooling) to "G"(postgraduate) for different family incomes indicated in the curves.

The experimental procedure had logistic regression as predictive technique and the comparison was performed with single-tailed t-test on a 10-fold cross-validation setting on paired test sets. The predictors (features) were partitioned into 3 groups as planned: Infrastructure, SEC information and Education. The performance metric was the Area Under the ROC curve (AUC_ROC) widely applied for assessing binary classifiers in domains such as medicine, telecommunications, artificial intelligence etc.

The results show that both the groups of features of parents' and teachers' education and of socio-economic-cultural information are more influential than the group composed of infrastructure features with statistical significance of 0.05.

Some research found in the literature argue that there is a high correlation among the predictors and that there could be causality in SEC information influencing the Education predictors. We have shown that Education predictors have a positive effect on the students' performance no matter the family income. Nevertheless, much more analyses have to be made in that sense.

Furthermore, ensemble of predictors in general achieve higher improvement in performance with the increase of complementarity among their modules [31]. That suggests that the higher increase of AUC in the combination of SEC and education features versus the combination of SEC and infrastructure features might be related to the smaller correlation of education compared to infrastructure both in relation to SEC features. This needs to be further investigated. It is also important to extend the analyses presented here for 2018 to several years to verify if the results found hold across time. We are carrying out the research and the preliminary results show that the same behavior holds for the previous 9 years as well.

It is important that experts in education and policy makers collaborate in this research to help improve the Domain-Driven Data mining approach by embedding their expertise in the solution development.

6. REFERENCES

- [1] Enem. *INEP*, 2020.
- [2] Microdados. *INEP*, 2020.
- [3] P. J. Adeodato. Data mining solution for assessing brazilian secondary school quality based on enem and census data. In *Proc. 13 CONTECSI*, pages 2658–2679, 2016.
- [4] P. J. L. Adeodato, F. C. Pereira, and R. F. O. Neto. Optimal categorical attribute transformation for granularity change in relational databases for binary decision problems in educational data mining. *CoRR*, abs/1702.08745, 2017.
- [5] S. Barretto. Políticas de formação docente para a educação básica no Brasil embates contemporâneos. *Revista Brasileira de Educação*, 20:679 – 701, 2015.
- [6] R. J. Bolton and D. J. Hand. Statistical fraud detection: A review. *Statistical science*, pages 235–249, 2002.
- [7] J. Brophy and T. Good. Teacher behavior and student achievement in m. witrock (ed.), the third handbook of research on teaching (pp. 328–375), 1986.
- [8] M. M. Burke. Improving parental involvement: Training special education advocates. *Journal of Disability Policy Studies*, 23(4):225–234, 2013.
- [9] D. Card. The causal effect of education on earnings. In *Handbook of labor economics*, volume 3, pages 1801–1863. Elsevier, 1999.
- [10] M. Carnoy, T. Khavenson, L. Costa, I. Fonseca, and L. Marotta. Is brazilian education improving? a comparative foray using pisa and saeb brazil test scores. *A Comparative Foray Using PISA and SAEB Brazil Test Scores (December 16, 2014)*. *Higher School of Economics Research Paper No. WP BRP*, 22, 2014.
- [11] A. Chevalier, C. Harmon, V. O’Sullivan, and I. Walker. The impact of parental income and education on the schooling of their children. *IZA Journal of Labor Economics*, 2(1):8, 2013.
- [12] A. Chudgar, T. Luschei, and L. Fagioli. Constructing socio-economic status measures using the trends in international mathematics and science study data. *East Lansing: Michigan State University*, 2012.
- [13] J. S. Coleman, E. Campbell, C. Hobson, J. McPartland, A. Mood, F. Weinfeld, et al. Equality of educational opportunity study. *Washington, DC: United States Department of Health, Education, and Welfare*, 1966.
- [14] L. Darling-Hammond. Teacher quality and student achievement. *Education policy analysis archives*, 8:1, 2000.
- [15] P. E. Davis-Kean. The influence of parent education and family income on child achievement: the indirect role of parental expectations and the home environment. *Journal of family psychology*, 19(2):294, 2005.
- [16] P. Dolton, O. Marcenaro, R. d. Vries, and P.-W. She. Global teacher status index 2018. 2018.
- [17] T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.
- [18] D. D. Goldhaber and D. J. Brewer. Evaluating the effect of teacher degree level on educational performance. 1996.
- [19] C. A. T. Gomes and M. R. T. Duarte. School infrastructure and socioeconomic status in Brazil. *Sociology and Anthropology*, 5(7):522–532, 2017.
- [20] J. Han, J. Pei, and M. Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.
- [21] R. A. Johnson, D. W. Wichern, et al. *Applied multivariate statistical analysis*, volume 5. Prentice hall Upper Saddle River, NJ, 2002.
- [22] H. Jones. More education, better jobs? a critical review of CCTs and Brazil’s Bolsa Família programme for long-term poverty reduction. *Social Policy and Society*, 15(3):465–478, 2016.
- [23] C. Longbing. Introduction to domain driven data mining. In *Data Mining for Business Applications*, pages 3–10. Springer, 2009.
- [24] P. Lundborg, M. Nordin, and D. O. Rooth. The intergenerational transmission of human capital: the role of skills and health. *Journal of Population Economics*, 31(4):1035–1065, 2018.
- [25] F. J. Murillo and M. Román. School infrastructure and resources do matter: analysis of the incidence of school resources on the performance of Latin American students. *School effectiveness and school improvement*, 22(1):29–50, 2011.
- [26] J. J. S. Neto, G. R. De Jesus, C. A. Karino, and D. F. De Andrade. Uma escala para medir a infraestrutura escolar. *Estudos em Avaliação Educacional*, 24(54):78–99, 2013.
- [27] OECD. *Rethinking Quality Assurance for Higher Education in Brazil*. 2018.
- [28] R. L. Silva Filho and P. J. Adeodato. Data mining solution for assessing the secondary school students of Brazilian federal institutes. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 574–579. IEEE, 2019.
- [29] B. Stauss, F. Nordin, and C. Kowalkowski. Solutions offerings: a critical review and reconceptualisation. *Journal of Service Management*, 2010.
- [30] R. TRAVITZKI. *ENEM: limites e possibilidades do Exame Nacional do Ensino Médio enquanto indicador de qualidade escolar. 2013*. PhD thesis, Tese (Doutorado em Educação)—Universidade de São Paulo, São Paulo, 2013.
- [31] D. H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

A Procrastination Index for Online Learning Based on Assignment Start Time

Lalitha Agnihotri¹, Ryan S. Baker², Steve Stalzer¹

McGraw Hill Education¹, University of Pennsylvania²

Lalitha.Agnihotri@mheducation.com, rybakera@upenn.edu, Steve.Stalzer@mheducation.com

ABSTRACT

Despite decades of evidence on the impacts of procrastination on learner outcomes, the educational data mining community has procrastinated in applying measures of procrastination based on learner behavior. We advance a new measure of habitual procrastination within online learning, the *Procrastination Index*, which represents a learner's degree of procrastinating in when they start learning assignments (rather than when they complete assignments), relative to other learners within the same assignment (recognizing that different assignments may need different amounts of time). We apply this measure to data from over 100,000 students in 3,700 course sections from a large online learning platform. We find that students who habitually delay starting assignments have 21 times the risk of failing their courses than students who start on time. The result of this work is a straightforward and reliable Procrastination Index that generalizes across multiple academic disciplines, takes the individual features of assignments into account, is a strong predictor of academic performance, and provides an early signal to enable educators to design appropriate interventions for at-risk students.

Keywords

Procrastination, educational data mining, at-risk prediction

1. INTRODUCTION

Everyone procrastinates sometimes – even psychological researchers studying procrastination [8]. Despite procrastination's near-universality as a phenomenon, though, understanding is still incomplete as to what the full effects of procrastination are, where it emerges from, and how it can be combatted.

The relationship between procrastination and academic performance has been studied extensively. A meta-analysis by van Eerde [20] found that students who procrastinate generally receive worse course grades, a result seen in online learning environments as well [7, 12, 23]. On the other hand, other researchers have found evidence that students who procrastinate experience less stress and have better health than students who do not procrastinate [19].

A range of procrastination behaviors appear to be associated with poorer outcomes. Although procrastination has been defined rather

broadly as “the tendency to postpone an activity under one's control to the last possible minute, or even not to perform it at all” [6], most studies of procrastination involve homework or studying. However, even procrastinating on accessing course materials is associated with worse course outcomes [1]. Several factors appear to be associated with the decision to procrastinate, from anxiety and depression [3] (though see [19] for contrasting evidence), to self-handicapping [20], to poor self-regulation [12] or a lack of scaffolding for self-regulation [16].

However, there are key limitations to past research on procrastination. Importantly, most published papers on the topic assess procrastination through self-report measures [11,18]. While these self-report measures correlate to behavioral measures such as whether the student hands in assignments late and total time spent, the correlation is moderate, in the -0.2 to -0.3 range [20]. Furthermore, this is not quite the same as identifying actual procrastination – delaying in starting or working on an assignment. For instance, a student could start early, work hard throughout, but still turn in a difficult assignment late. It is also conceivable that some students may think they are procrastinating more than other students when they are not. Correspondingly, some highly successful students may procrastinate, starting at the last minute, and still turn in high-quality work on time. These students may not see themselves as procrastinators. Therefore, in this paper we attempt to hone more closely in on procrastination as a behavior, using learning system data to see when students start an assignment as well as when they turn it in, following recent work in the EDM community using log data to study procrastination [i.e. 4, 9, 13].

In the remainder of this paper, we begin by offering an operational definition of procrastination at the level of a learning task and then aggregating it to the level of a learner. We study the properties of procrastination according to this definition, and then investigate the empirical relationship between procrastination and academic performance. We embed this into an analysis of the probabilistic risk associated with different levels of procrastination according to our definition. Finally, we present linear and logistic regression models that use procrastination on tasks to predict students' final grade and whether they will pass or fail the course, as a method for applying this paper's findings into prediction-based interventions.

2. METHODS

We used two datasets for the study, Alpha and Beta, that were derived from the online learning system Connect, a web-based learning system actively used by approximately 6000 higher education institutions worldwide. Students use Connect to read a course text and complete assignments. Instructors can compose assignments from a question bank as well as creating their own assignments. Both instructor-created and question bank assignments can be auto-graded. Connect records assignment start and end time, and the grade. Dataset Alpha is a heterogeneous

Lalitha Agnihotri, Ryan Baker and Steve Stalzer "A Procrastination Index for Online Learning Based on Assignment Start Time" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 550 - 554

dataset spanning multiple institutions. Alpha consists of 2,666,617 assignment submissions by 102,506 students on the platform during the Fall 2018 semester. The assignments span 3,681 courses, 42 disciplines, and 3,681 instructors at 1,216 institutions. Although the platform is used internationally, we restricted the analysis to US institutions to limit regional issues, policy differences in data use, and possible cultural differences in procrastination. The students submitted about 112,025 unique assignments in various courses. The courses are set up by instructors and differ in terms of course length, the number of homework assignments (the sample was restricted to courses with at least 10 assignments), and what percentage of the overall course grade is made up of the assignments on the Connect platform.

Dataset Beta, a more homogeneous data set from a single institution, also contained the final course grade for each student. The dataset, collected in the 2018/2019 academic year, consists of 98,201 assignment submissions on 5,986 assignments by 1,022 unique students in 298 sections of 37 courses in 28 disciplines. Many students were included in more than one course for a total of 3758 student-sections. The courses are designed with a regular spacing of assignments, four per week in each of eight weeks, for a total of approximately 32 assignments per course. In these courses, assignments on Connect are worth 80% of the course grade.

3. OPERATIONAL DEFINITIONS

3.1 Task Procrastination

All procrastination is delay, but not all delay is procrastination [15]. The central concept in procrastination is task delay – i.e. delaying in starting or completing a task that needs to be completed to accomplish some goal. When the student considers when to start an assignment, the student must decide, explicitly or implicitly, how much time they will need and, therefore, when they should start. An error in estimating this correctly places the student at risk of a poor grade. As a first step, let us postulate that for each assignment there is a threshold time to start the assignment, τ_t , a point after which we cannot reasonably expect most students to perform well on the assignment due at time τ_d . Note that this is a simplifying assumption: student knowledge of the topic and general ability likely varies, causing the true threshold start time to vary between students for a given assignment [cf. 10].

Consider two scenarios. In the first, a student begins a task at time τ_s before the threshold time τ_t and is therefore likely to complete the task and complete it well.



Figure 1 "Safe Zone" for starting an assignment

In the second scenario, a student begins a task after the threshold time, and is not likely to obtain a good grade on the assignment.

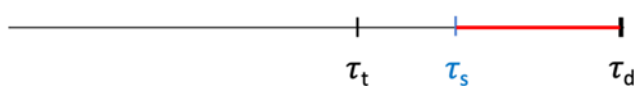


Figure 2 Case when start time is after threshold time

But how does one find τ_t ? Or in other words, how do we assess task procrastination for a specific task given that the time needed to complete will vary from task to task? We can answer this question by considering the average z-scores derived based on each assignment rather than the absolute scores. Figure 3 shows the average grade z-scores students achieved based on when they start an assignment. By the 60th percentile, the score is below the mean performance on the homeworks. Near the 75th percentile, the score has dropped to 10% less than the mean and the decrease accelerates.

Based on these findings, we can heuristically set the threshold time τ_t for an assignment to be κ^{75}_s , the start time at which 75% of students have started the assignment. Setting κ^{75}_s as the threshold time, we can assign each student and each assignment a Boolean value to indicate whether the student started their assignment early enough or whether they procrastinated. A value of 0 means the student started their assignment early enough that we can say they did not procrastinate. A value of 1 means the student procrastinated. In other words, if τ_s is before κ^{75}_s , the student started on time. If τ_s is after κ^{75}_s , the student procrastinated. In the unfortunate special case where more than a quarter of students start after the due date, seen in approximately a quarter of assignments, we set τ_t to 0 -- starting after the due date is by definition procrastination, since no one can complete an assignment in less than 0 seconds.

$$\kappa^{75}_s = \kappa^{75}_s \text{ if } \kappa^{75}_s \leq \tau_d$$

$$\tau_d, \text{ if } \kappa^{75}_s > \tau_d$$

Task procrastination is then defined as follows. It is set to 0 if the start time is before the fourth quartile threshold κ^{75}_s as defined above. It is set to 1 if the start time is after this point or if no start time exists (the student never started the assignment)

$$P = 0 \text{ if } \tau_t < \kappa^{75}_s$$

$$P = 1 \text{ if } \tau_t > \kappa^{75}_s, \text{ or } \tau_t \text{ is null}$$

3.2 Learner Procrastination

We can now use this assessment of Task Procrastination t as the basis for creating a Learner Procrastination Index (PI). For example, the following array represents a student S1 and their procrastination pattern (again, 1 represents procrastination and 0 represents not procrastinating). Take a hypothetical student, Chris. Chris started the first two assignments on time, and procrastinated on the remaining ones, until beginning the final assignment on time.

$$P_{\text{Chris}} = [0; 0; 1; 1; 1; 1; 1; 1; 1; 0]$$

From this, we compute Chris's Procrastination Index (PI) as the percentage of 1s on a scale from 0 to 1, 0.7 based on the above.

$$PI = \text{mean}(P_{\text{Chris}})$$

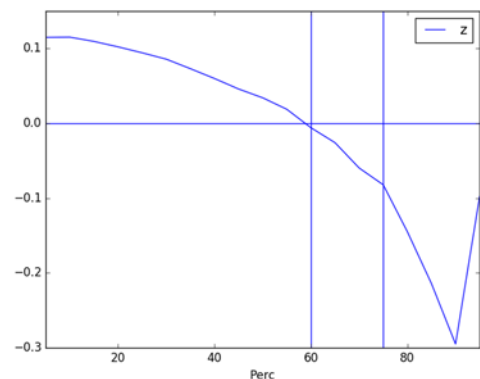


Figure 3 Starting Percentile vs. Z-Scored Grade

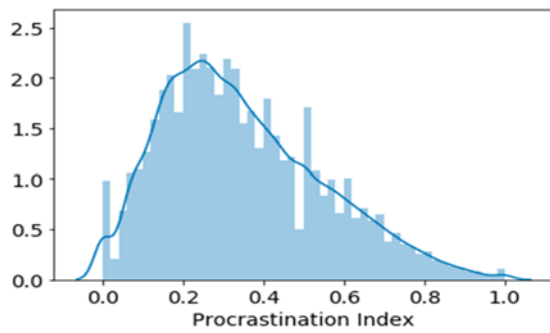


Figure 4 Histogram of PI for the 100k+ students

A PI of 0 means that student began every assignment on time. A PI of 1 means that the student procrastinated on every assignment. Figure 4 represents the distribution of the Procrastination Index for the over 100K students in Dataset Alpha.

4. Analysis of Procrastination, Performance, and Outcomes

With this operational definition of procrastination, we can now examine the relationship between procrastination and performance, shown in Figures 5 and 6. Figure 5 shows the average of the score on assignments in Connect for the dataset Alpha and figure 6 shows the average of the final grade on the course for dataset Beta. The average course grade declines as the Procrastination Index increases. The Pearson correlation between Procrastination Index and grade was found to be -0.67 and -0.69 for the datasets Alpha and Beta respectively, $p < 0.001$ for both datasets. It is worth noting that these correlations are double to triple the magnitude of the correlations to grades previously reported for self-report measures of procrastination ($r = -0.2$ to -0.29 ; [i.e. 20]). Furthermore, as Figure 6 shows, the relationship is fairly consistent. Students who procrastinate under 5% average an A grade; students who procrastinate under 20% of the time have above a B average. Students who procrastinate under half the time receive more Bs and As than Cs. As the graph shows, there is a relatively steep drop-off in grade around a PI of 50%. Students who procrastinate 95% of the time tend to obtain a D or F.

In the remainder of this section, we will analyze the difference in course grades between students who frequently procrastinate (high Procrastination Index; “high PI”) and students who procrastinate less often (“low PI”). These cut-offs are somewhat arbitrary, and we set them using course grades; although this creates some circularity, the resultant analysis is correlational rather than causal and therefore should be considered descriptive in nature.

Given the sharp drop-off in grades seen at a Procrastination Index of around 50% (see Figure 6), we can consider students who procrastinate more than half of the time to have high procrastination. There is not quite as clear a cut-off for low procrastination, but given that 20% marks a point where students tend to get Bs or better, we can consider 20% a cutoff for low procrastination. To create a group of students with medium PI for analysis, we chose PI between 0.3 and 0.4 to have values evenly between low PI and high PI while having a gap between groups.

Figure 7 shows the probability distribution function of Dataset Alpha for performance for different PI groups. Students with a high PI (red) are distributed at the lower end of the performance range. Students with low PI (green) tend to have higher performance and have low probability of obtaining an average score of under 60%.

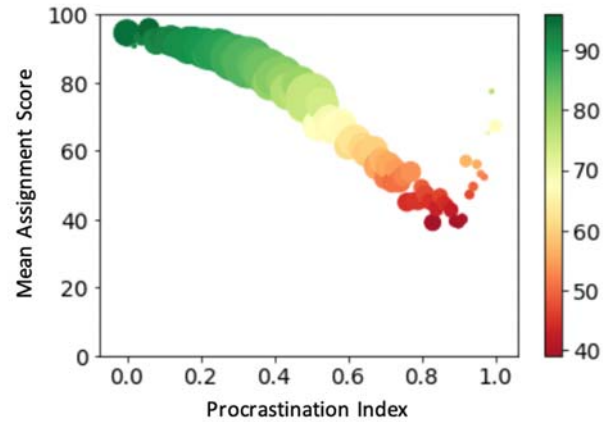


Figure 5 PI vs. Mean Assignment Score

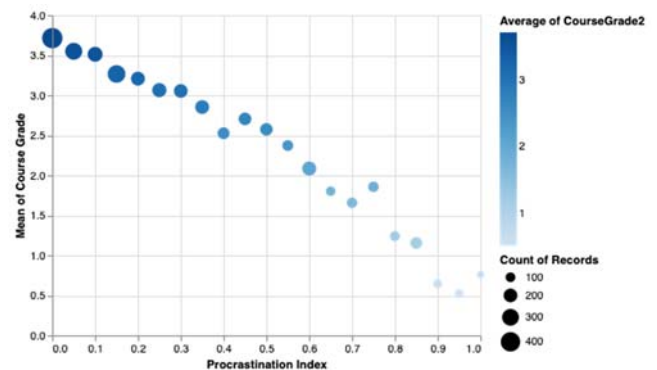


Figure 6 PI vs Mean Course Grade (A: 4, B: 3, C: 2, D: 1, F/W:0)

4.1 Procrastination and the Risk of Failure

Based on these categorizations, we can study the degree to which students with high and low PI are at different levels of risk of failing a course. For Dataset Alpha, we classify a student as passing if they obtain a grade of 70 or higher for the course. For Dataset Beta, we have obtained the actual final grades from the university. A/B/C is defined as pass; D and all other grades (F and a never-completed “incomplete” or withdraw) are treated as a failing grade.

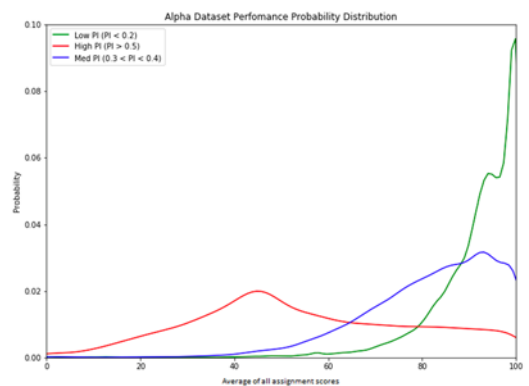


Figure 7 Performance (mean assignment score) Distribution for Different PI ranges for Dataset Alpha

Within Dataset Alpha, high PI students fail 71.5% of the time, while low PI students fail 3.4% of the time ($RR = 21$). Specifically, we compute the risk ratio (RR) for the likelihood that a student with a high PI will fail the course, compared to the likelihood that low

PI student will fail. We also compare the score distributions with Cliff's Delta, a measure of the degree of overlap between two distributions. Cliff's Delta scales from -1 to +1, where +1 and -1 indicate no overlap (in the two directions), 0 indicates total overlap, and values in between indicate partial overlap.

Dataset Beta, high PI students fail 54.6% of the time, while low PI students fail 1.1% of the time (RR=50). The Cliff's Delta is 0.82 for dataset Alpha is 0.82 (median score 91 vs. 54) and 0.77 for Beta (median score 92 vs. 67), indicating very little overlap between the two distributions. Nearly every student in the low PI group outscores every student in the high PI group.

4.2 Procrastination as a Predictor of Outcomes

In this section, we investigate PI as a potential predictor of final score and course outcome, using linear regression to predict the final score and logistic regression model to predict if a student would pass or fail the course, with a training-test split. We find that PI can be used to predict the final score, with $R^2 = 0.45$ for both datasets, and RMSE of 15 (Alpha)/ 17 (Beta) grade points. Logistic regression obtains a successful AUC ROC of 0.86 for both datasets. Even if we vary the cut-off for task procrastination, considering the 50th, 60th, 75th, 85th, and 95th percentile, and re-fit the model, model performance remains high. As Table 1 shows, the models maintain reasonably high AUC ROC across thresholds, with moderately higher AUC ROC with procrastination cut-offs from the 75th to 95th percentile of time. However, it is probably not useful for intervention to select a threshold where 95% of students have already started the assignment. Note that the recall values in this table do not fit the intuition that recall should go up for lower thresholds; this is because the threshold is at the level of individual assignments, whereas the logistic regression model sets a second cut-off at the level of students across assignments.

Table 1 Performance of logistic regression models that use different start time thresholds for procrastination.

Threshold Percentile	Average Precision	Average Recall	AUC ROC	Precision of Students who fail	Recall of Students who fail
50	69	59	0.8	55	22
60	72	63	0.83	60	31
75	76	67	0.86	65	40
85	78	69	0.87	68	44
95	78	69	0.87	68	43

5. DISCUSSION AND CONCLUSIONS

Though there has been considerable work on procrastination over the last decades, much of this work has looked at self-report measures or submission time. In this paper, we consider when students start assignments, relative to other students' work on the same assignment, which can function across contexts and can be aggregated across a course. Our aggregation, termed the Procrastination Index, is correlated with not only score within the Connect platform, but with the overall grade on the course, and can predict student grades, achieving double to triple the correlation to student outcomes seen for prior self-report measures [i.e. 20].

We can use early detection of procrastination to message students and to help them develop good habits. Even students who are performing well, but frequently procrastinate, may benefit from developing better habits – procrastination may become a bigger problem for these students when they reach more difficult material. Finishing tasks just in time can make sense in specific cases – but if students develop a general strategy of procrastinating, it may misserve them later [2]. Several interventions may be successful at

helping students to work effectively. [21] have recently published a meta-analysis of different interventions designed to reduce procrastination, looking at which type of intervention leads to the strongest reduction. They investigated interventions involving self-regulated learning strategies (including time management), cognitive-behavioral therapy, and assertiveness training. They found that cognitive-behavioral therapy led to significantly less procrastination, and that assertiveness training actually led to significantly more procrastination. However, all of the interventions investigated in [21] were intensive. By contrast, [2] has proposed a way for students to offer their own deadlines to avoid a last minute rush to complete, leading to improved grades. In an automatic system, we can envision a system enabling students to suggest deadlines or presenting additional deadlines (for, say, a milestone that represents completing half the homework) to help them break down the task and reduce procrastination. It may also be possible to create automated interventions inspired by cognitive-behavioral therapy, although it is unclear whether they will work as well as the full therapeutic approach.

It remains to be seen what interventions are most effective at reducing procrastination and improving outcomes in a scalable fashion. As with other domains such as help-seeking [cf. 17], the relationship between procrastination and outcomes is probably not fully causal and it may be possible to reduce procrastination without improving outcomes. Finding the right intervention(s) to improve outcomes will be beneficial not only in improving outcomes but also in understanding whether – and how – procrastination has causal impacts on learning. More generally, a fuller understanding of procrastination may help us to better alleviate its impacts. Do students procrastinate as a habit or is it an ongoing error in their estimation of their time demands? What role do boredom and lack of engagement play? Better understanding the answers to these questions may ultimately lead to redesign of courses and/or assignments to better keep students engaged in their learning in a steady fashion throughout the semester.

In this paper, we have proposed a way to identify procrastination in students based on their interactions with an online learning system, that accounts for start time relative to other students. The PI indicator seems to generalize well across many different class sections, subject areas, and disciplines. We have been able to apply it to over a hundred thousand student scores in the Connect learning platform as well as with around 3,700 students at a specific institution with their final course grades. The correlation of Procrastination Index to the outcome in the course is around -0.7. The PI on a course can be used in a linear regression model to predict the final score, achieving an R^2 of 0.45, substantially higher than the predictive power of self-report measures of procrastination. For predicting pass or fail using a logistic regression model based solely on procrastination, we are able to achieve an area under the ROC curve of 0.86. We plan to use this research to improve our products – targeting content that is often procrastinated on for improvements -- and develop ways to nudge students to work more effectively and finish their tasks earlier. If we, as a field, stop procrastinating on this important issue, the impact on our students may be profound.

ACKNOWLEDGMENTS

We would like to thank the developers of the Connect platform for supporting our data collection efforts, and Alfred Essa, for initial leadership of this research while at McGraw Hill.

6. REFERENCES

- [1] Agnihotri, L., Essa, A., and Baker, R. 2017. Impact of Student Choice of Content Adoption Delay on Course Outcomes. *Proceedings of the 7th International Conference on Learning Analytics and Knowledge*, 16-20.
- [2] Ariely, D. 2010. *Predictably Irrational: The Hidden Forces That Shape Our Decisions*.
- [3] Beswick, G., Rothblum, E.D., and Mann, L. 1988. Psychological antecedents of student procrastination. *Australian Psychologist* 23, 2 (1988), 207-217.
- [4] Cerezo, R., Esteban, M., Sánchez-Santillán, M., and Núñez, J. C. Procrastinating behavior in computer-based learning environments to predict performance: a case study in Moodle. *Frontiers in Psychology*, 8, 1403.
- [5] Edwards, S. H., Snyder, J., Pérez-Quinones, M. A., Allevato, A., Kim, D., and Tretola, B. 2009. Comparing effective and ineffective behaviors of student programmers. *Proceedings of the International Computing Education Research Workshop*, 3-14.
- [6] Gafni, R. and Geri, N. 2010. Time management: Procrastination tendency in individual and collaborative tasks. *Interdisciplinary Journal of Information, Knowledge, and Management*, 5, 115-125.
- [7] Goda, Y., Yamada, M., Kato, H., Matsuda, T., Saito, Y., and Miyagawa, H. (2015). Procrastination and other learning behavioral types in e-learning and their relationship with learning outcomes. *Learning and Individual Differences*, 37, 72-80.
- [8] Lay, C. H. 1986. At last, my research article on procrastination. *Journal of Research in Personality*, 20(4), 474-495.
- [9] Levy, Y. and Ramim, M. 2012. A Study of Online Exams Procrastination Using Data Analytics Techniques. *Interdisciplinary Journal of E-Learning and Learning Objects*, 8(1), 97-113.
- [10] Ma, Y., Agnihotri, L., Baker, R., and Mojarad, S. 2016. Effect of student ability and question difficulty on duration. *Proceedings of the 9th International Conference on Educational Data Mining*, 135-142.
- [11] McCloskey, J. and Scielzo, S. 2015. Finally!: The Development and Validation of the Academic Procrastination Scale. Unpublished Manuscript. Retrieved 12/24/2019 from https://www.researchgate.net/profile/Shannon_Scielzo/publication/273259879_Finally_The_Development_and_Validation_of_the_Academic_Procrastination_Scale/links/54fcfb3d0cf20700c5e9c735.pdf
- [12] Park, S.W. and Sperling, R.A.. 2012. Academic procrastinators and their self-regulation. *Psychology*, 3, 1-12.
- [13] Park, J., Yu, R., Rodriguez, F., Baker, R., Smyth, P., and Warschauer, M. 2018. Understanding Student Procrastination via Mixture Models. *Proceedings of the 11th International Conference on Educational Data Mining*.
- [14] Parson, D. E., and Seidel, A. 2014. Mining Student Time Management Patterns in Programming Projects. *Proceedings of the International Conference on Frontiers in Computer Science & Computer Engineering Education*, 21-24.
- [15] Pychyl, T.A. 2009. Active procrastination: Thoughts on oxymorons. *Psychology Today*.
- [16] Rakes, G. C., and Dunn, K. E. 2010. The Impact of Online Graduate Students' Motivation and Self-Regulation on Academic Procrastination, *Journal of Interactive Online Learning*, 9 (1), 78-93.
- [17] Roll, I., Aleven, V., McLaren, B. M., and Koedinger, K. R. 2011. Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction*, 21 (2), 267-280.
- [18] Solomon, L. J. and Rothblum, E. D. 1984. Academic procrastination: Frequency and cognitive-behavioral correlates. *Journal of Counseling Psychology*, 31, 503-509.
- [19] Tice, D.M. and Baumeister, R.F. 1997. Longitudinal study of procrastination, performance, stress, and health: The costs and benefits of dawdling. *Psychological Science*, 8 (6), 454-458.
- [20] van Eerde, W. 2003. A meta-analytically derived nomological network of procrastination. *Personality and Individual Differences*, 35 (6), 1401-1418.
- [21] van Eerde, W. and Klingsieck, K. 2018. Overcoming Procrastination? A Meta-Analysis of Intervention Studies. *Educational Research Review*, 25, 73-85.
- [22] Wang, Z. and Englander, F. 2010. A cross-disciplinary perspective on explaining student performance in introductory statistics--what is the relative impact of procrastination? *College Student Journal*, 44(2), 458-472.
- [23] You, J. W. 2015. Examining the effect of academic procrastination on achievement using LMS data in e-learning. *Journal of Educational Technology & Society*, 18(3), 64.

Automated Assessment of Computer Science Competencies from Student Programs with Gaussian Process Regression

Bitakram¹, Hamoon Azizolsoltani¹, Wookhee Min¹, Eric Wiebe¹, Anam Navied¹,
Bradford Mott¹, Kristy Elizabeth Boyer², James Lester¹

¹North Carolina State University, Raleigh, North Carolina
{bakram, wmin, hazizso, wiebe, anavied, bwmott, lester}@ncsu.edu

²University of Florida, Gainesville, Florida
keboyer@ufl.edu

ABSTRACT

Recent years have seen a growing interest in learning analytics for computer science education. The significant growth of computer science enrollments coupled with the labor-intensive nature of human assessment has fueled the demand for automated assessment of student programs. Effective automated assessment tools can bridge the gap between the demand for support and restricted instructional resources by providing adaptive formative and summative feedback. Following an evidence-centered assessment design approach, we have devised an automated assessment framework for middle grades computational thinking. We report on an evaluation comparing regression models including ridge, lasso, support vector, and Gaussian process regression models utilizing a structural n -gram feature set to infer scores for students' programs. The results show that Gaussian process regression outperforms other regression models with respect to mean squared error and adjusted coefficient of determination. They also show that the framework provides a promising approach with regard to dealing robustly with noise to effectively model student computer science competencies.

Keywords

Competency Modeling, Automated Program Assessment, Computer Science Education

1. INTRODUCTION

Computer science (CS) has become a foundational skill for students to thrive in a digital economy [14, 28]. To prepare students for future studies and science and technology professions, it is essential to ensure that they acquire robust CS competencies. A key strategy for developing CS competencies is through programming. However, learning how to program is challenging for novices [12, 13]. Hence, novice programmers need significant scaffolding to support understanding and effective use of CS concepts. Effective assessment of students' understanding of essential CS concepts is an important step in providing students with appropriate scaffolding and feedback [11, 17]. Because the growth in demand for CS education has outstripped the supply of qualified teachers, providing every student with individualized, high-quality, and

timely support and feedback is challenging. Effective automated assessment can guide adaptive formative and summative feedback to support effective CS education.

In order to provide students and their instructors with reliable automated assessments, we follow a hypothesis-driven learning analytic approach [4] based on Evidence-Centered Assessment Design (ECD) [20] to assess students' competencies in CS concepts as demonstrated in their programs. Following this approach, we first identify CS concepts that students need to master in order to solve a particular computational thinking-focused problem with a block-based programming interface embedded in the ENGAGE game-based learning environment (Figure 1). We then collect log data from students' interactions with the game. Content area experts then analyze the structured log data as evidence of knowledge (or lack thereof) of target CS concepts. Deriving evidence from students' proposed solutions, we assess their mastery of identified CS concepts, such as creating appropriate algorithms and programs, and appropriate usage of computer science constructs, such as loops and conditionals. We encode programs as structural n -grams to extract essential structural and semantic information within them. Finally, we utilize regression models including ridge, lasso, support vector regression (SVR), and Gaussian process regression (GPR) models on the generated feature set to infer students' competencies for knowledge of CS concepts and practices. We utilize GPR models to handle the remaining noise in the dataset. Evaluation results suggest that the models accurately model students' CS competencies and are robust to noise.

2. RELATED WORK

Two primary approaches have been explored for assessing text-based programs: dynamic and static assessment [5, 15]. In dynamic assessment, programs are executed against pre-defined test data to determine their correctness. Evaluation metrics include successful compilation, consideration of security threats, correct outcome, and efficiency metrics such as CPU runtime and clock time [15, 16, 25]. In contrast, static assessments are capable of assessing programs that are not necessarily complete. To perform a static assessment, an intermediate representation of the program needs to be generated from the source code. Examples of intermediate representations are textual representations, abstract syntax trees, control flow graphs, and program dependence graphs. After forming the intermediate representation, the representation is analyzed for its correctness, efficiency, and quality [26]. Although block-based programming differs from text-based programming in syntax and visual representation, they can both be transformed into the same intermediate representation. Therefore, the techniques used for

Bitakram, Hamoon Azizolsoltani, Wookhee Min, Eric Wiebe, Bradford Mott, Anam Navied, Kristy Elizabeth Boyer and James Lester "Automated Assessment of Computer Science Competencies from Student Programs with Gaussian Process Regression" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 555 - 560

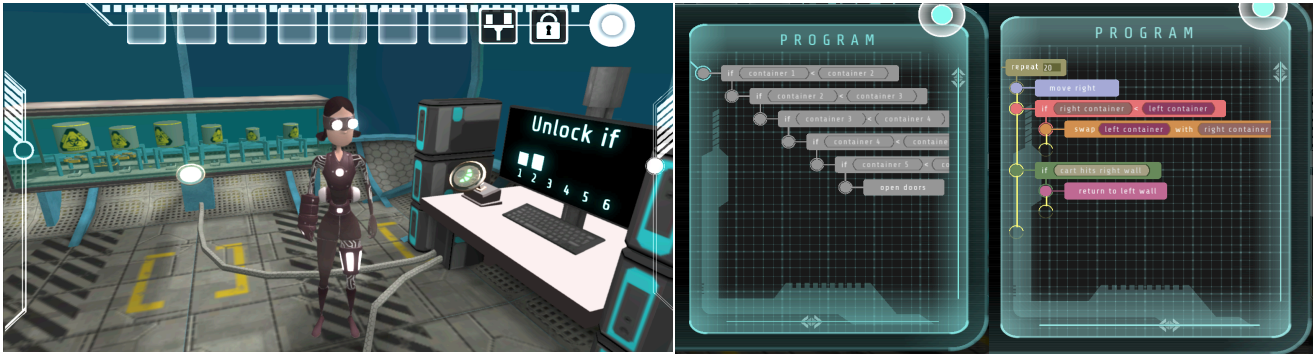


Figure 1. ENGAGE game-based learning environment. (Left) The bubble sort task in the game-based learning environment. (Right) Program for the bubble sort task: the read-only code for opening the door and an example of a correct implementation of the bubble sort written by a student.

assessing one type of programming from this representation can be readily adapted to assess the other type of programming [9, 10].

Limited previous work has focused on assessing students' understanding of underlying CS concepts from their programs. In this paper, we propose an automated assessment framework designed following a hypothesis-driven learning analytics approach to assess students' programs based on their mastery of underlying CS concepts for the particular problem at hand. We use the underlying CS concepts to label students' programs with their grades. We then transform students' programs to a feature set containing salient features that can serve as evidence for students' proficiency of this underlying CS concepts. Utilizing the labeled data and the extracted feature set predictive models can identify students' mastery of CS concepts.

3. ENGAGE LEARNING ENVIRONMENT

To collect data on middle grade students' programming trajectories, we conducted a study with a game-based learning environment, ENGAGE, that is designed to teach CS to middle school students [2, 18]. The game features a rich, immersive 3D storyworld (Figure 1), in which students play the role of a specialist who is sent to investigate an underwater research facility that has lost communication with the outside world through suspicious activities of a rogue scientist. In the learning environment, students navigate through a series of interconnected rooms by solving a set of computational challenges. Each of the challenges can be solved either by programming devices within the room or interacting with devices in reference to their pre-written programs. Students program the devices with a visual block-based programming language interface (Figure 1, Right) [1, 19].

In this work, we focus on students' problem-solving approaches within a specific level of the game where students write a bubble sort algorithm to order a set of containers (Figure 1). This room has two devices: a containment device that holds six randomly positioned containers and a lock device that opens only when the containers are sorted in the increasing order. The player can exit the room through a door by correctly implementing bubble sort and executing the lock program when the containers are sorted. The lock has a pre-written program that will check the positions of containers and opens if they are in the correct position. The containment device provides students with the necessary blocks for implementing a bubble sort algorithm using a small robotic cart inside the device's protective housing. Students can choose from 6

types of readily available blocks to write their program. A sample correct solution for this challenge is shown in Figure 1 (right). Students need to test the correctness of their program in two steps. First, they need to run the bubble sort device to sort the containers. Second, they need to run the open lock program which checks if the containers are sorted and opens the door accordingly.

4. METHODOLOGY

We utilize supervised learning to assess students' programs. The supervised learning approach consists of three primary steps. First, we label the training dataset in accordance with a rubric designed based on essential CS constructs. Second, we extract features from students' submitted program snapshots that represent their understanding of CS constructs. This is accomplished with a novel approach that encodes students' programs in terms of structural n -grams. Third, we create models induced from the structural n -gram-based feature set to infer students' scores. In this study, we utilize a variety of regression models including linear, ridge, lasso, SVR, and GPR models to predict students' programs' scores.

4.1 Data Annotation

We use evidence-centered assessment design (ECD) to create a rubric for labeling students' programs [22]. Following an ECD approach, we identify explicit learning outcomes and measures to inform our rubric [7]. The relevant CS concepts are identified and used to develop the specifications of a rubric to assess students' proficiency of identified CS concepts. Student actions during the learning task are used as evidence for predicting mastery of the identified CS concepts [24]. Following this approach, we design a rubric that utilizes evidence rules specific to the bubble sort challenge in the ENGAGE game-based learning environment. We use this rubric to manually label students' programs [3, 4].

As students interacted with the learning environment, all of their interactions with the game were logged. For this study, we collected data from five classrooms across three schools in the United States. We collected data from 69 students' interactions with the bubble sort challenge in the game-based learning environment, for a total of 1,570 programs that we used as the training dataset. In this rubric, the range of possible scores is between 0 to 22. To validate the labeling process, two human annotators with a computer science background separately labeled 20% of the entire submissions, achieving an inter-rater agreement of 0.856 using Cohen's kappa [8]. Before tagging the remainder of the corpus, all instances that were tagged differently were discussed. Afterwards,

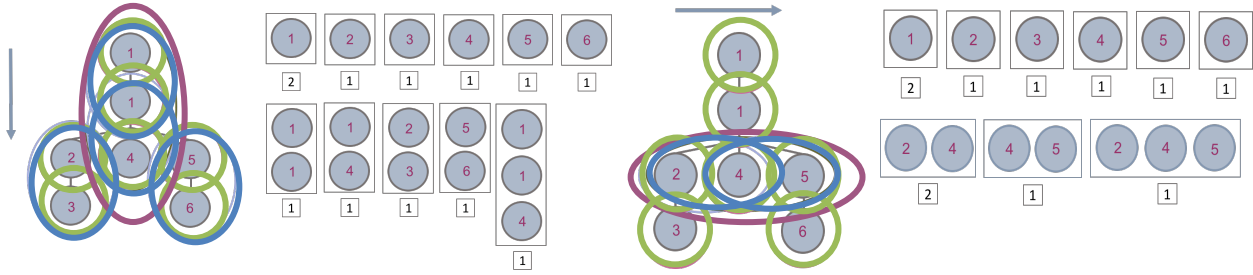


Figure 2: AST generated from a sample program submitted for the bubble sort challenge and its hierarchical and ordinal n -gram encoding. (Left) An AST and its partial *hierarchical* unigrams, bigrams, and 3-grams marked by green, blue and purple ovals respectively on the left and the partial feature set generated from hierarchical n -gram encoding of the AST along with feature-level frequencies on the right. (Right) An AST and its *ordinal* unigrams, bigrams, and 3-grams marked by green, blue and purple ovals respectively on the left and the partial feature set generated from partial ordinal n -gram encoding of the AST along with feature-level frequencies on the right.

one annotator tagged the remainder of the dataset. The annotated dataset is used as the corpus for our prediction task where the assigned scores serve as the ground-truth for our data corpus.

4.2 Feature Engineering

In order to infer students' scores based on their submitted programs, we extract structural features that are representative of the semantic information in students' code. To extract features that capture essential structural and semantic information stored within the programs, we perform a structural n -gram encoding of the programs' abstract syntax trees (ASTs). Since the programs are stored as programming snapshots, we first apply an intermediate transformation from programming snapshots to their corresponding ASTs [23]. We then encode the generated ASTs into their corresponding structural n -grams. After generating the ASTs from their corresponding programs, we use a structural n -gram encoding to capture essential structural information characterized within the programs. Two important structural information types in ASTs are *hierarchical* and *ordinal*. The hierarchical information encodes what blocks are nested under another (i.e., a vertical relationship in AST), and the ordinal information encodes the placement order of blocks (i.e., horizontal relationship in AST) that are nested under the same parent node. We extract n -grams with varying lengths of n to capture the most fine-grained structural information present in an AST. We repeat the n -gram encoding process separately for hierarchical feature extraction and ordinal feature extraction. We then merge the two feature sets together to build the final feature set containing both hierarchical and ordinal n -gram encodings corresponding to each program keeping only one copy of the generated unigrams. The occurrence of similar n -grams for n values more than one (unigrams) in both hierarchical and vertical encodings demonstrate presence of different structures in the AST and thus, both will be preserved.

Figure 2 shows an AST generated from a sample program along with its partial hierarchical (left) and ordinal (right) n -gram encoding. In Figure 2 (left), each colored circle shows the hierarchical (top to bottom) n -gram encoding of a specific n . In this example, we have hierarchical encoding of n -grams of size one (green ovals), two (blue ovals) and three (the purple ovals). The frequency values for each n -gram encoded feature are shown beside the AST. All of the other n -gram feature values are zero since they are not in this AST. Figure 2 (right) shows the same sample AST with its ordinal (left to right) n -gram encoding. In this example, we

have an ordinal encoding of n -grams of size one (pink ovals), two (purple ovals) and three (the green ovals). Similar to Figure 2 (left), the frequency values for each n -gram feature is shown besides its corresponding AST in Figure 2 (right).

4.3 Inferring Program Scores

We trained a variety of regression models on the structural n -gram-encoded features to infer the scores of students' programs. In particular, we used linear regression as the baseline regression model, and four additional regression models: ridge, lasso, support SVR, GPR models. Ridge and lasso regressions are characterized by their utilization of L1 and L2 regularization, respectively, which address overfitting and variance issues in comparison to linear regression. We use SVR and GPR models since their utilization of kernels makes them a suitable candidate for datasets similar to ours where the number of features is relatively high compared to the number of data points. Finally, we utilize GPR to handle the noise resulting from the subjective nature of human grading [6, 27]. To infer students' program grades using the n -gram encoded feature set (we set the maximum n to 4 for hierarchical n -grams and 3 for ordinal n -grams in this work), we use a 5-fold cross-validation approach to tune the hyperparameters of ridge, lasso, and SVR regressions, and to identify the appropriate kernel for the GPR. After the hyperparameter optimization process is complete, we use 10-fold cross-validation to train and evaluate each regression model.

4.3.1 Linear Regression

Linear regression is a simple regression approach that works under the assumption that there is a linear relationship between features and the predicted value. The results of applying a 10-fold cross-validation evaluation on the n -gram encoded feature set resulted in a Mean Squared Error (MSE) of $3.03\text{E}+24$ and an R-squared of $1.19\text{E}-23$. The high MSE value reported by the linear model trained with our feature set is understandable since the high number of features in our dataset dramatically increases the complexity of the model, which in turn causes overfitting of linear regression-based predictive models to the training data.

4.3.2 Ridge Regression

To reduce the variance error, ridge regression includes a penalty term in the optimization. We used the set $[0.05, 0.1, 0.5, 1.0, 10]$ to tune the value for λ , the penalty coefficient, and found $\lambda=10$ to be

the best value for our regression task. Applying ridge regression on our dataset resulted in an MSE of 5.24 and an R-squared of 0.81. We can see that ridge regression considerably outperformed standard linear regression with respect to both MSE and R-squared.

4.3.3 Lasso Regression

Unlike ridge regression, lasso regression includes a penalty term that allows the optimization process to shrink weights to zero if necessary. As a result, lasso regression can reduce overfitting as well as perform feature selection. We used the set [0.05, 0.1, 0.5, 1.0, 10] as in ridge regression to tune the value for λ and found $\lambda=0.05$ to be the best value for λ . Utilizing lasso regression resulted in an MSE of 6.30 and an R-squared of 0.77, which also outperformed standard linear regression models with respect to both MSE and R-squared.

4.3.4 Support Vector Regression

Support vector regression (SVR) use kernels to transform data from a non-linearly separable space to a linearly separable space. For our regression task, we explored with linear, polynomial, and radial basis function (RBF) kernels. For each kernel, we tuned the hyperparameters of penalty parameter (C), epsilon, and kernel coefficient (gamma). For polynomial kernels, we also tuned the parameter of the kernel projection ($coef0$) and degree hyperparameters. Utilizing 5-fold cross-validation, we found the polynomial kernel with a degree of four to be the best kernel for our dataset. Also, the grid search identified $C=100$, $coef0=1$, degree= 4, epsilon=1, and gamma= 0.001 as the best parameters for this kernel. Incorporating the SVR model resulted in an MSE of 5.09 and an R-squared of 0.82. SVR performed better than both ridge and lasso regressions in terms of MSE and R-squared. This could be due to the fact that kernel methods perform effectively on datasets with a feature set that is relatively large compared to the size of the dataset.

4.3.5 Gaussian Process Regression

GPRs provide an analytically tractable solution for regression problems with an infinite or uncountable set of considered basis functions [21]. We hypothesize that the GPR will outperform other regression techniques due to its capability of handling noise and its propriety for our dataset. To search the optimal kernel for GPR models, we cross-validated the model for radial basis functions (RBF), rational quadratic, and Matern kernels, and we found RBF to perform the best on our dataset. To find the optimal set of hyperparameters and prior parameters of the GPR, we follow the process of maximizing the probability of observing data given hyperparameters of the process (i.e., marginal likelihood). In this work, we use a limited-memory BFGS optimization technique to maximize the log marginal likelihood conditioned on the length vectors and the noise level of the kernels.

Applying GPR resulted in an MSE of 1.71, and an R-squared of 0.94. GPR performed significantly better than other regression models. Not only is GPR a kernel-based model similar to SVR, but by adding an additional noise kernel it can also account for the potential noise in our dataset. As a result, it is expected that the GPR model outperformed other models in our prediction task. Results of applying each of the regression models on the structural n -gram encoded feature set is shown in Table 1.

Table 1. Average predictive performance of regression models trained with the structural n -gram feature set

Regression	MSE	R ²
Linear	3.03E+24	1.19E-23
Ridge	5.24	0.81
Lasso	6.30	0.77
SVR	5.09	0.82
GPR	1.71	0.94

5. CONCLUSION

Rapidly growing interest in computer science education and students' need for guided practice of CS concepts have created a significant need for accurate and effective automated assessment. In this work, we proposed a novel structural n -gram encoding scheme to extract important structural and semantic information from students' programs. The n -gram encoding approach, coupled with data labeled using the ECD-based rubric enables our assessment framework to model evidence from programs that are representative of students' mastery of identified CS. We apply a variety of regression models on the n -gram encoded feature set to infer students' program scores. The results of our prediction demonstrate the effectiveness of the n -gram encoded feature set in capturing important semantic and structural information in students' programs. All regression models performed better than the baseline model, linear regression. Furthermore, GPR outperformed other models in terms of both mean squared and R-Squared errors. This confirms expectations since GPR models can handle noisy data and are particularly efficient for datasets in which the number of features is particularly high relative to the number of data points. Our automated CS competency assessment framework can be generalized to assess any well-structured programs in learning environments that present students with well-structured programming assignments. Furthermore, the ECD approach can facilitate rubric design and assessment for non-expert CS teachers while providing them with automated assessment of students' programs.

Several directions for future work are promising. First, it will be important to expand the assessment framework to accommodate more open-ended programming assignments. Second, information from successive submission of students can be extracted to analyze students' patterns of developing CS competencies.

6. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation under Grants DRL-1640141. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

7. REFERENCES

- [1] Akram, B. 2019. Assessment of Students' Computer Science Focal Knowledge, Skills, and Abilities in Game-Based Learning Environments. Ph.D. Dissertation. North Carolina State University, Raleigh, NC.
- [2] Akram, B., Min, W., Wiebe, E., Mott, B., Boyer, K., and Lester, J. 2018. Improving Stealth Assessment in Game-based Learning with LSTM-based Analytics. In *Proceedings of the 11th International Conference on Educational Data Mining*, 208–218.
- [3] Akram, B., Min, W., Wiebe, E., Mott, B., Boyer, K., and Lester, J. 2019. Assessing Middle School Students' Computational Thinking Through Programming Trajectory Analysis. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 1269–1269.
- [4] Akram, B., Min, W., Wiebe, E., Navied, A., Mott, B., Boyer, K., and Lester, J. 2020. A conceptual assessment framework for k-12 computer science rubric design. In *Proceedings of the 51th ACM Technical Symposium on Computer Science Education*, 1328.
- [5] Ala-Mutka, K. 2005. A Survey of Automated Assessment Approaches for Programming Assignments. *Computer Science Education* 15, 2, 83–102.
- [6] Amershi, S. and Conati, C. 2009. Combining Unsupervised and Supervised Classification to Build User Models for Exploratory Learning Environments. *Journal of Educational Data Mining, Article 1*, 1, 18–71.
- [7] Brennan, K., and Resnick, M. 2012. New frameworks for studying and assessing the development of computational thinking. *Annual American Educational Research Association meeting, Vancouver, BC, Canada*, 1–25.
- [8] Cohen, J. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1, 37–46.
- [9] Emerson, A., Rodríguez, F., Mott, B., Smith, A., Min, W., Boyer, K., Smith, C., Wiebe, E., and Lester, J. 2019. Predicting Early and Often: Predictive Student Modeling for Block-Based Programming Environments. In *Proceedings of the 12th Conference on Educational Data Mining*, 39–48.
- [10] Emerson, A., Smith, A., Rodríguez, F., Wiebe, E., Mott, B., Boyer, K., and Lester, J. 2020. Cluster-based analysis of novice coding misconceptions in block-based programming. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 825–831.
- [11] Fields, D., Giang, M., and Kafai, Y. 2014. Programming in the wild: trends in youth computational participation in the online scratch community. In *Proceedings of the 9th workshop in primary and secondary computing education*, ACM, 2–11.
- [12] Grover, S., and Basu, S. 2017. Measuring Student Learning in Introductory Block-Based Programming. In *Proceedings of the 48th ACM SIGCSE Technical Symposium on Computer Science Education*, 267–272.
- [13] Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., and Stamper, J. 2017. A Framework for Using Hypothesis-Driven Approaches to Support Data-Driven Learning Analytics in Measuring Computational Thinking in Block-Based Programming Environments. *ACM Transactions on Computing Education* 17, 3, 14.
- [14] Hansen, A., Dwyer, H., Iveland, A., Talesfore, M., Wright, L., Harlow, D., and Franklin, D. 2017. Assessing Children's Understanding of the Work of Computer Scientists: The Draw-a-Computer-Scientist Test. In *Proceedings of the 48th ACM SIGCSE technical symposium on computer science education*, 279–284.
- [15] Ihantola, P., Ahoniemi, T., Karavirta, V., and Seppälä, O. 2010. Review of Recent Systems for Automatic Assessment of Programming Assignments. In *Proceedings of the 10th Koli calling international conference on computing education research*, 86–93.
- [16] Lajis, A., Baharudin, S., Kadir, D., Ralim, N., Nasir, H., and Aziz, N. 2018. A Review of Techniques in Automatic Programming Assessment for Practical Skill Test. *Journal of Telecommunication, Electronic and Computer Engineering* 10, 2, 109–113.
- [17] Meerbaum-Salant, O., Armoni, M., and Ben-Ario, M. 2013. Learning computer science concepts with scratch. *Computer Science Education* 23, 3, 239–364.
- [18] Min, W., Frankosky, M., Mott, B., Rowe, J., Wiebe, E., Boyer, K., and Lester, J. 2015. DeepStealth: Leveraging Deep Learning Models for Stealth Assessment in Game-based Learning Environments. In *Proceedings of the 17th International Conference on Artificial Intelligence in Education*, 277–286.
- [19] Min, W., Frankosky, M., Mott, B., Wiebe, E., Boyer, K., and Lester, J. 2017. Inducing Stealth Assessors from Game Interaction Data. In *Proceedings of the 19th International Conference on Artificial Intelligence in Education*, 212–223.
- [20] Mislavy, R., Haertel, G., Riconscente, M., Rutstein, D., and Ziker, C. 2017. Evidence-Centered Assessment Design. In *Assessing Model-Based Reasoning using Evidence-Centered Design*. SpringerBriefs in Statistics, 19–24.
- [21] Rasmussen, C.. 2004. Gaussian Processes in machine learning. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 63–71.
- [22] Rupp, A., Pearson, K., Sweet, S., Crawford, A., Levy, I., Fay, D., Kunze, K., Cisco, M., Mislavy, R., and Pearson, J. 2012. Putting ECD into Practice: The Interplay of Theory and Data in Evidence Models within a Digital Learning Environment. *Journal of Educational Data Mining* 4, 1, 49–110.
- [23] Shamsi, F., and Elnagar, A. 2012. An Intelligent Assessment Tool for Students' Java Submissions in Introductory Programming Courses. *Journal of Intelligent Learning Systems and Applications* 04, 01, 59–69.
- [24] Snow, E., Haertel, G., Fulkerson, D. and Feng, M. 2010. Leveraging evidence-centered assessment design in large-scale and formative assessment practices. In *Proceedings of the 2010 Annual Meeting of the National Council on Measurement in Education (NCME)*.
- [25] Taherkhani, A., and Malmi, L. 2013. Beacon- and Schema-Based Method for Recognizing Algorithms from Students' Source Code. *Journal of Educational Data Mining* 5, 2, 69–101.
- [26] Wang, T., Su, X., Wang, Y., and Ma, P. 2007. Semantic

- similarity-based grading of student programs.
Information and Software Technology 49, 2, 99–107.
- [27] Zen, K., Iskandar, D., and Linang O. 2011. Using Latent Semantic Analysis for automated grading programming assignments. In *International Conference on Semantic Technology and Information Retrieval*, 82–88.
- [28] 2016. K-12 Computer Science Framework. Retrieved August 25, 2018 from <http://www.k12cs.org>.

Educational Data Mining and Personalized Support in Online Introductory Physics Courses

Farook Al-Shamali
Athabasca University
farooka@athabascau.ca

Hongxin Yan
University of Eastern Finland
hongya@student.uef.fi

Sabine Graf
Athabasca University
sabineg@athabascau.ca

Fuhua Lin
Athabasca University
oscarl@athabascau.ca

ABSTRACT

Physics has always been a challenging subject for many students. Research also shows a gap between instructional goals and actual student learning in introductory physics courses. This study focuses on two online first-year courses that cover classical mechanics of the physics curriculum at an open university in Canada. Each of the two courses is developed around a textbook and includes a locally created study guide enriched with animated videos, dynamic diagrams, and interactive exercises. This study aims at introducing a simple feature to provide physics students with personalization based on their background knowledge and at examining students' interactions with the online course materials. Relevant educational data are compiled using checkpoint quizzes, self-reflection questionnaires, examinations, and log data collected through the learning management system (Moodle). In addition, peer faculty feedback is collected. Positive correlations are expected between regular learning behavior and engagement in personalized support and students' performance on examinations.

Keywords

Data mining, Learning analytics, Learning management system, Moodle, Introductory physics, Distance education, Online learning, Personalized support, Learning behaviours.

1. INTRODUCTION

Despite its significance as a foundation for modern technological achievements, physics is perceived as a challenging subject by many students. In 1987, a prominent physicist, Richard Feynman (1918 – 1988), suggested “that physics shouldn’t be taught in high school because most of the teachers lacked a passion for the subject” [1]. Researchers at the time also pointed out an alarming gap between expected learning outcomes and actual student learning in introductory physics courses [2]. This old problem called for a reconsideration of the traditional approach to teach this important subject.

The argument surrounding physics education is especially relevant to the distance education (DE) model, which is witnessing a period of accelerated evolution, powered by advancements in digital technology. Despite challenges linked to the nature of DE, the flexible presentation format of online courses breaks some of the traditional barriers and allows for new possibilities. This leads to the question about effective instructional design features in introductory physics courses that cater to all students and provide successful online experiences [3-5].

Farook Al-Shamali, Hongxin Yan, Sabine Graf and Fuhua Lin "Educational Data Mining and Personalized Support in Online Introductory Physics Courses" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 561 - 564

An online course delivered through a learning management system (LMS) can provide a multitude of data and information related to students' interactions with the course materials. Knowledge obtained from mining and analyzing available data, in combination with plug-in adaptive learning systems, can be used to guide individual students to study more effectively and to improve the quality and presentation of the course materials [6-8]. For instance, recent studies suggest that one of the common (and apparently less productive) students' practices in studying physics involves the “cramming” of relatively large quantities of the subject matter during short periods preceding exams [5,9]. Also, Imhof et al. indicated a “negative relationship between prior knowledge test score and predicted learning progress” in physics modules [3]. Even though the investigated courses cater to adult learners, not all students can effectively acquire online learning abilities [10]. Such observations highlight how personalized and adaptive learning are potentially effective concepts in the design of online physics courses.

A major advantage of the traditional face-to-face (F2F) educational model is the direct student-teacher interaction, which permits the instructor to make timely adjustments to the subject matter and teaching style to ensure better students' engagement. The assumption here is that the instructor is sufficiently flexible to make the required accommodations, and the size of the classroom is reasonably small so that accommodating individual students becomes practical. In an online class, however, students interact less with a dedicated teacher but more with the LMS and the course materials. This is especially true in the asynchronous delivery model, where course content (typically) consists of rigid learning resources developed with the “one-size-fits-all” teaching concept. Such a delivery format does not take into consideration the “individual differences, personal needs and personal development” of all students [7].

Chaw and Tang found that students' use of the LMS is correlated with the service quality it provides [11]. Also, the quality of an online course should be enhanced when instructors are equipped with effective learning analytics and data mining tools [12,13]. In particular, proper utilization of educational data promises to facilitate effective personalized learning in online courses, including personalized feedback and recommendations for extra learning materials [8,14-18]. Such individualized support is particularly important in physics courses where conceptual understanding is typically constructed vertically using scaffoldings provided by essential mathematical tools. Therefore, physics students are expected to appreciate personalized learning environments that evaluate their progress, fill individual knowledge gaps, and sharpen specific math skills if needed [19,12].

Learning analytics (LA) and educational data mining (EDM) have been used for a range of applications, including personalized learning [20]. In this paper, we introduce a work-in-progress

research project that uses LA and EDM to examine students' interactions with the course materials in two online physics courses. More specifically, the project introduces a simple and practical adaptive feedback module that can be easily integrated into the LMS. It provides a level of personalization based on students' background knowledge directed toward reducing the knowledge gap among a diverse student group.

2. PHYSICS COURSES

This study focuses on two first-year physics courses offered online (through Moodle) at Athabasca University in Canada. The first course is an algebra-based introductory physics and covers conventional topics in classical mechanics. It is considered among the top 50 high enrollment courses at the university, with effective annual registrations that exceed 400 students. The second course is physics for scientists and engineers, which is the calculus-based version of the first course. Both courses share a mandatory home lab component [21].

The textbook is an essential educational resource in a typical physics course. However, traditionally, the textbook is written with the conventional classroom in mind. Therefore, in DE, the study guide becomes an important component that guides the student through different learning activities and course assessments. In particular, the study guides for the two courses are designed to complement the textbook and provide additional reading (and audiovisual) materials related to each unit and lab experiment (see Figure 1). An important component of the study guide consists of detailed solutions to physics problems related to each unit in the course.

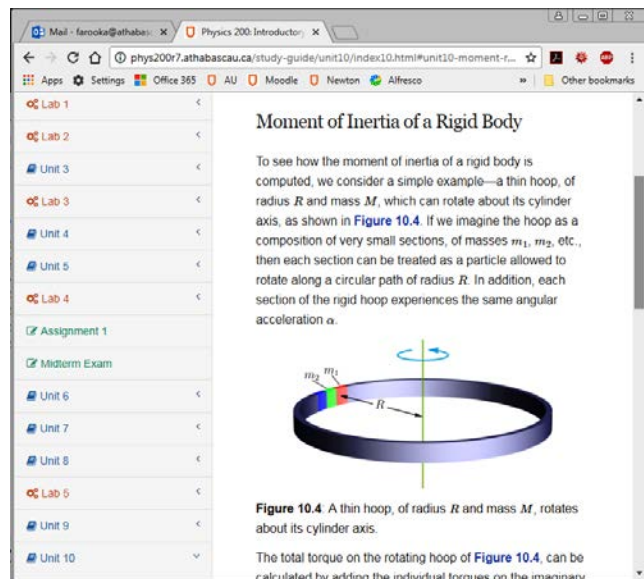


Figure 1. Snapshot from the introductory physics study guide.

Recent revisions of both investigated courses involved the production of an online study guide written in HTML with MathJax scripting of the LaTeX code. The improved version of the study guide is enriched with dynamic and interactive diagrams created using Mathematica (<https://www.wolfram.com/mathematica/>). This is in addition to the free simulations of the PhET Project (<https://phet.colorado.edu/>). Students are expected to benefit from the interactivity and the real-time visualization of interactions between position, velocity and acceleration, especially in two and three dimensions. This includes the kinematics and dynamics of

projectiles, circular motions, collisions, etc. Some of the interactive diagrams are complex enough to be considered virtual labs that simulate real-life situations.

The course development team constructed a website for the study guide that is accessible through the course homepage on Moodle and supporting responsive (mobile optimized) features. This is in addition to the textbook, which is accessible as an eTextbook through a separate link. One of the courses uses an open educational resource (OER textbook). The new design approach to the study guides received positive feedback from peer faculty members. However, even though some design considerations are integrated into the course for collecting feedback, our knowledge of students' interaction with the course content is limited.

3. RESEARCH Questions

In this study, we investigate the effectiveness of automated personalized support provided to students at specific milestones in two online physics courses. More specifically, the study addresses the following research questions:

- Is there a correlation between the academic performance of individual students and their response and behavior concerning the adaptive feedback module?
- How do learning behaviour and study patterns influence students' overall academic performance?
- What course elements are most effective regarding the adaptive feedback module?

4. RESEARCH PLAN

Relevant educational data are compiled using checkpoint quizzes, students' self-reflection questionnaire, course assessment results, and log data collected through the LMS (Moodle). In addition, peer faculty feedback will be collected.

4.1 Personalization through checkpoint quizzes

For the proposed personalization feature, the online study guide for each course is divided into five sections covering the main topics in each syllabus: kinematics, dynamics, energy & momentum, gravity & rotational motion, and elasticity & equilibrium.

Before starting a new section, a student is encouraged to complete a multiple-choice checkpoint quiz that is automatically marked by the LMS. The optional quiz is used as a checkpoint to assess the student's mastery of the topics in each section (see Figure 2). Based on the responses, the system may suggest the student proceeds to the next unit in the course or recommend a set of additional learning resources that may help strengthen the student's specific background concepts required by the upcoming topics. For example, a student who underachieved on the quiz questions related to rotational motion could be directed to a relevant video (such as <https://youtu.be/garegCgMxxg>) from the Khan Academy (<https://www.khanacademy.org/>), the problem-solving examples created in the study guide, or a section of the textbook. Apparently, there is limited research on "the effectiveness of such actionable links on students' learning experience and success" as stated by Iraj et al. [22]. The authors also warned that most students appear to lack "feedback literacy" and may only respond to quality feedback. This research project aspires to provide an informative contribution in this regard by using Moodle Quiz module's overall feedback

feature, which can provide different feedback for a different level of quiz performance.

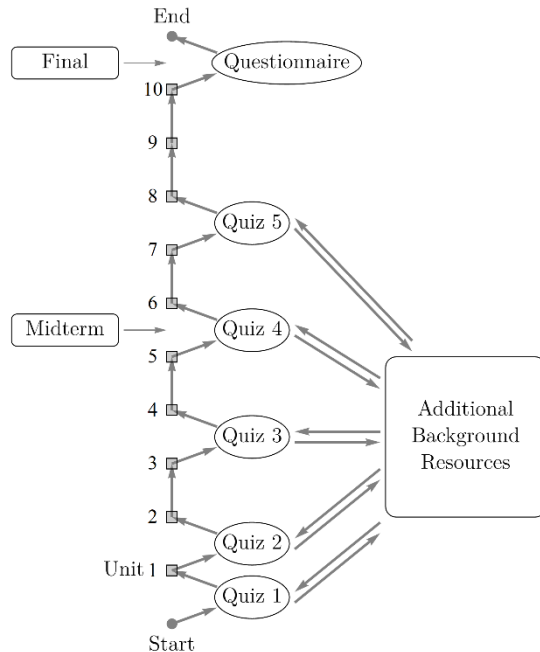


Figure 2. Learning path and checkpoints in the online physics course.

4.2 LMS Log Data

The LMS automatically records the date, time, and score of each checkpoint quiz and keeps track of other learning activities, including recommended learning objects accessed by individual students. When combined with the timing and marks achieved on assignments, lab reports and exams, each student's progress throughout the course materials can be highlighted. Such information allows us to look for patterns of effective learning behaviour and explore the effectiveness of the proposed adaptive feature on the student's academic achievement in the course.

4.3 Student Self-Reflection Questionnaire and Faculty Feedback

The effectiveness of the course content and design, in addition to the student's learning behaviour, is also gauged through a self-reflection questionnaire completed by the student toward the end of the course. The questionnaire is conducted online and consists of a mix of multiple-choice and written response questions. The collected data provides self-reflection by the students on their study behaviour, feedback on the proposed adaptive feature, convenience of course design, and effectiveness of course content, especially the interactive exercises and dynamic diagrams. Also, we will solicit qualitative assessment and feedback from instructors and tutors about the efficiency and effectiveness of the system through interviews.

4.4 Data Analysis

Based on the results of the first three checkpoint quizzes (see Figure 2), we group students by score quartile (students who perform below 25%; students with a score between 25% and 50%; students with a score between the 50% and 75%; and the students who score above 75%.) We then follow the learning behaviour of each group and their performance on the midterm examination. The fifth group of students who choose to skip the checkpoint quizzes,

continuing from one unit to the next, can act as a reference group. A similar analysis is repeated for data collected during the second half of the course.

To compare the use of recommended learning resources across quartiles, we compute the mean number of resources accessed for each quartile. We hypothesize that students with higher exam scores tend to engage more seriously with feedback and follow a more regular study pattern (i.e., suggested study schedule) than those with lower exam scores. We will see if the response to feedback between student groups is significant at the $p < 0.05$ level for different quizzes. The findings can be used to detect struggling students since they are less likely to use exercise for study purposes. Also, we conjecture that the students with the lowest grades have the lowest score on checkpoint quizzes and follow a more random study pattern. Their learning behaviour may be guessing or viewing hints in an attempt to build a catalog of correct answers, rather than actively using their knowledge to correctly address their knowledge gaps and adopt a more productive learning behaviour.

5. CONCLUSION

Students' interaction with the course materials, combined with their use of personalization through checkpoint quizzes, the results of self-reflection questionnaires, and peer faculty feedback, are analyzed in association with student's performance on assignments, lab reports and exams. The purpose is to look for educationally meaningful information regarding effective personalized feedback, successful learning behaviour, and good aspects of instructional design. An extension to this project would involve investigating the impact of personalizing the study schedule on the issue of procrastination and student attrition in online physics courses.

6. REFERENCES

- [1] Hewitt, P. G., Ed. 2015. *Conceptual physics* (12th Ed.). Pearson Addison-Wesley, Petersburg, FL, USA.
- [2] McDermott, L. C. 1990. Millikan Lecture 1990: What we teach and what is learned—Closing the gap. *American Journal of Physics*, 59, 4, 301–315.
- [3] Imhof, C., Bergamin, P., Moser, I., and Holthaus, M. 2018. Implementation of an Adaptive Instructional Design for a Physics Module in a Learning Management System. In *Proceedings of the 15th International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2018)*, 69–78. https://www.celda-conf.org/wp-content/uploads/2019/04/CELDA_2018.pdf
- [4] Kortemeyer, G. 2016. Work Habits of Students in Traditional and Online Sections of an Introductory Physics Course: A Case Study. *Journal of Science Education and Technology*, 25, 697–703.
- [5] Kortemeyer, G. 2019. It's All in the Data - But What is It? Learning Analytics and Data Mining of Multimedia Physics Courses. *International Journal of Physics & Chemistry Education*, 11, 1, 13–17.
- [6] Dietz-Uhler, B. and Hurn, J. E. 2013. Using Learning Analytics to Predict (and Improve) Student Success: A Faculty Perspective. *Journal of Interactive Online Learning*, 12, 1, 17–26.
- [7] Peng, H., Ma, S., and Spector, J. M. 2019. Personalized adaptive learning: an emerging pedagogical approach enabled by a smart learning environment. *Smart Learning*

Environments, 6, 9. <https://doi.org/10.1186/s40561-019-0089-y>

- [8] Khosravi, H., Sadiq, S., and Gasevic, D. 2020. Development and adoption of an adaptive learning system. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, (83-88). <https://doi.org/10.1145/3375462.3375532>
- [9] Suhonen, S. J. and Tiili, J. A. 2015. Students' Online Activity on a Fully Online Introductory Physics Mechanics Course. In *Proceeding of the 43rd Annual SEFI Conference*, <https://www.sefi.be/wp-content/uploads/2017/09/55889-SJ-SUHONEN.pdf>
- [10] Martinez, M. 2002. What Is Personalized Learning? The eLearning Developers' Journal, May 7, 1–8.
- [11] Chaw, L. Y. and Tang, C. M. 2018. What Makes Learning Management Systems Effective for Learning? *Journal of Educational Technology Systems*, 47, 2, 152–169.
- [12] CHEN, L. 2019. Enhancing Teaching with Effective Data Mining Protocols. *Journal of Educational Technology Systems*, 47, 4, 500–512.
- [13] Martin, F., Ndoeye, A., and Wilkins, P. 2016. Using Learning Analytics to Enhance Student Learning in Online Courses Based on Quality Matters Standards? *Journal of Educational Technology Systems*, 45, 2, 165–187.
- [14] Armstrong, and Fuhua, L. 2010. Modeling and Personalizing Curriculum Using Petri Nets. In *Proceedings of the 18th International Conference on Computers in Education*, 10-12.
- [15] Graf S., Kinshuk (2012) Personalized Learning. In: Seel N. M. (eds.) *Encyclopedia of the Sciences of Learning*. Springer, Boston, MA, 2592-2594.
- [16] Greller, W. and Drachsler H. 2012. Translating Learning into Numbers: A Generic Framework for Learning Analytics. *Educational Technology & Society*, 15, 3, 42–57.
- [17] El-Bishouty, M. M., Chang, T. -W., Lima, R., Thaha, M. B., Kinshuk, and Graf, S. 2015. Analyzing Learner Characteristics and Courses Based on Cognitive Abilities, Learning Styles, and Context. In Chang, M. and Li, Y. (eds.), *Smart Learning Environments, Lecture Notes in Educational Technology*, Springer-Verlag, Berlin Heidelberg, 3-25.
- [18] Roberts, L.D., Howell, J.A. and Seaman, K. 2017. Give Me a Customizable Dashboard: Personalized Learning Analytics Dashboards in Higher Education. *Tech Know Learn* 22, 317–333. <https://doi.org/10.1007/s10758-017-9316-1>
- [19] Bautista, G. 2012. The effects of personalized instruction on the academic achievement of students in physics. *International Journal of Arts and Sciences*, 5, 5, 573–583.
- [20] Baker, R. 2016. Using learning analytics in personalized learning. *Handbook on personalized learning for states, districts, and schools*, 165-174.
- [21] Al-Shamali, F., and Connors, M. 2010. Low-Cost Physics Home Laboratory. In Kennepohl, D. and Shaw, L. (eds.), *Accessible Elements: Teaching Science Online and at a Distance*, Published by AU Press, Athabasca University, 131-146.
- [22] Iraj, H., Fudge, A., Faulkner, M., Pardo, A., and Kovanović, V. 2020. Understanding Students' Engagement with Personalised Feedback Messages. In *Proceedings of the 10th International Conference on Learning Analytics and Knowledge (LAK '20)*, March 23–27. <https://doi.org/10.1145/3375462.3375527>

First Steps Towards NLP-based Formative Feedback to Improve Scientific Writing in Hebrew

Moriah Ariely
Weizmann Institute of Science
Rehovot, Israel
moriah.ariely@weizmann.ac.il

Tanya Nazaretsky
Weizmann Institute of Science
Rehovot, Israel
tanya.nazaretsky@weizmann.ac.il

Giora Alexandron
Weizmann Institute of Science
Rehovot, Israel
giora.alexandron@weizmann.ac.il

ABSTRACT

As scientific writing is an important 21st century skill, its development is a major goal in high school science education. Research shows that developing scientific writing skills requires frequent and tailored feedback, which teachers, who face large classes and limited time for personalized instruction, struggle to give. Natural Language Processing (NLP) technologies offer great promise to assist teachers in this process by automating some of the analysis. However, in Hebrew, the use of NLP in computer-supported writing instruction was until recently hindered by the lack of publicly available resources. In this paper, we present initial results from a study that aims to develop NLP-based techniques to assist teachers in providing personalized feedback in scientific writing in Hebrew, which might be applicable to other languages as well. We focus on writing inquiry reports in Biology, and specifically, on the task of automatically identifying whether the report contains a properly defined research question. This serves as a proof-of-concept of whether we can build a pipeline that identifies major components of the report and match them to a predefined grading rubric. To achieve this, we collected several hundreds of reports, annotated them according to a grading rubric to create a supervised data set, and built a machine-learning algorithm that uses NLP-based features. The results show that our model can accurately identify the research question or its absence. To the best of our knowledge, this is the first paper to report on the application of Hebrew NLP for formative assessment in K-12 science education.

Keywords

Scientific writing, Formative assessment, Natural Language Processing

1. INTRODUCTION

Writing is a critical 21st century skill, and a high level of writing proficiency is required to succeed in academia and workplaces [1]. In science, writing is one of the primary

means of communication in the scientific community and a crucial aspect of scientific literacy. Thus, developing writing skills has become a major educational goal in high school science education [11].

Numerous studies have shown that developing scientific writing skills among high school students poses considerable difficulties for both students and teachers [9, 15, 23]. A lot of this may be due to the lack of formative feedback, which is known to be essential for the development of these skills [17, 10]. Formative feedback aims to guide and improve students' learning by providing them with information about the gap between their current and the desired performance. In the context of formative feedback on scientific writing, it has been shown that in order to support students in improving the quality of their writing, the formative feedback needs to be personalized and specific [3, 14]. It should also provide applicable recommendations for improvement, and explanations as to why such improvements are needed [16].

Proper writing instruction demands a significant amount of time from teachers, for preparing materials, reading, editing, and providing feedback. The educational reality is that teachers are faced with large class sizes that limit their ability to find the necessary time to devote to this process, resulting in a considerable delay in the feedback that students receive, and in its quality [1]. Another challenge is designing guidance that motivates students to engage in substantial writing revisions. Consequently, revising written explanations based on personalized guidance rarely occurs in science classrooms [19].

Technology holds much promise for improving this process, by supporting teachers in providing formative assessment. Automated computer scoring systems are being developed in order to address the challenges of assessing students' writing (e.g., [22, 19, 21, 18, 12, 20]). Among these, automated essay scoring technologies can enhance both large scale assessment and classroom instruction [3], as they have many advantages in the fields of assessment and instruction including objectivity, standardization and efficiency [5]. However, these technologies were mostly employed for *summative*, rather than *formative*, purposes [21].

In addition, while automated supporting tools for revising texts on the micro-level (such as grammar and spelling) are well represented [18], tools that support the development of writing strategies including self-monitoring and improving

Moriah Ariely, Tanya Nazaretsky and Giora Alexandron "First Steps Towards NLP-based Formative Feedback to Improve Scientific Writing in Hebrew" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 565 - 568

macro-level of text quality (such as argumentative structure and rhetorical moves) are infrequent [18]. In the transition from essay scoring to writing instruction, feedback design is of great importance, since it is the primary means through which students can evaluate and improve their writing [17].

In Hebrew, the use of NLP in computer-supported writing instruction was until recently hindered by the lack of publicly available resources. Hebrew is a morphologically rich language (MRL). It is complex, characterized by a highly productive inflectional morphology, with prefixes, suffixes and circumfixes, and also nouns, adjectives and numerals inflections for number (amount) and gender [8]. Following recent developments in Hebrew NLP, the high-level goal of our research is developing NLP-based techniques to assist teachers in providing personalized feedback in scientific writing in Hebrew, which might be applicable to other languages as well.

The task that we study is writing scientific reports in inquiry projects in Biology. A fundamental component of the report is a well-formulated research question(s). Formulating research questions that can be answered empirically is one of the practices needed in order to become scientifically literate [11]. In fact, by ‘composing questions’, students attend to the main ideas and check if the content is understood [13]. Since the research questions are defined on the early stages of the project, failure to properly define them can have a long-term effect on the quality of the project and report.

With this rationale in mind, we focus on studying NLP-based means to provide personalized feedback on the quality of the questions that students define. A precondition for an automatic assessment of the quality of the research questions is detecting them automatically in the text. The identification of the research questions in students’ essays serves as a proof-of-concept (POC) for learning a formative assessment grading scheme for major components of the report.

Our work is the first step towards NLP-based tools that will support K-12 science educators in teaching and assessing scientific writing in Hebrew. To the best of our knowledge, there is no published work on NLP-based formative assessment in Hebrew, and this research has the potential to pioneer this exciting domain.

2. METHOD AND RESULTS

This section describes the experimental setup, how the data was collected and annotated, the NLP pipeline and features, the machine learning algorithm, and the results.

2.1 Research context

Over 20,000 high-school students in Israel major in biology each year [4]. The Israeli Biology curriculum includes an inquiry project that constitutes 30% of the final grade [7]. It is conducted collaboratively in groups of 2-3 students. The students conduct an inquiry on a biological issue, ask research questions, design and carry out an experiment, collect data, and analyze it. Students are required to document their work in a scientific report. Within this process, the writing task was reported by teachers and students to

be the most challenging part [6]. It is an iterative process, which often takes up to 10 iterations to complete.

2.2 Data Collection

The data include 705 scientific reports, collected from 520 student groups that belong to 33 classes.

The reports are submitted in Hebrew as Word documents. In the first phase of the project the Introduction part, which is where the research question should be defined, was separated from the rest of the text. The Introduction typically consists of 2-5 pages. Well-written introduction section should contain the following discourse categories:

- Biological process.
- Research question. One research question if the work is submitted by two students, two research questions if the work is submitted by a group of three students.
- Research hypothesis.
- Description of the organism.

Following is an example of a typical well-written research question: "Our research question is how does alcohol concentration influence cell respiration rate in yeasts".

2.3 Creating a supervised data set

In order to create a supervised data set that can be used as an input to the machine learning algorithm, we annotated students’ texts. The goal of the annotation was to mark the segments of the texts that represent the aforementioned four discourse categories. The relevant parts of the texts were encoded with <tagname> and </tagname> tags that preceded/succeeded the relevant segments. For example, each research question was preceded with an <rq> tag, and succeeded by an </rq> tag, which were inserted into the text. The annotation was performed at a sentence or multiple sentences level. Each sentence was labeled with at most one discourse category. Our annotation scheme does not allow overlapping of the categories, but the same category may appear multiple times (e.g., two different research questions). We note that the majority of the sentences do not belong to any category and are not labeled at all.

The process was conducted by two domain experts (including one of the authors). The experts first created a grading rubric and then tagged the texts accordingly. In the first stage of the annotation process, both judges worked together to create a protocol for detecting the discourse elements in the text. Next, they worked independently to label 147 texts (from 44 student groups that belong to 6 classes), and the resulting labels were discussed until disagreements were resolved. Finally, additional 56 texts were labeled by one of the experts.

To create a training and test sets we chosen randomly one report from each student group, so the chosen reports represent different stages of report readiness. This means that some of the reports do not contain research questions at all and some research questions are ill defined. In total, the data set includes 100 texts containing 5513 sentences and 197 research questions.

2.4 Research Question identification

We consider the task of research question identification as a sentence-level classification task. Each sentence is classified as a research question or not. The data set was divided into training and test sets, as presented in Table 1.

Table 1: A summary of the annotated data.

	Number of texts	Number of sentences	Number of research questions
Training set	70	4013	139
Test set	30	1500	58
Total	100	5513	197

One of the challenges is that the data set is highly imbalanced. The ratio between examples in the minority class (research question sentences) and the majority class (non-research question sentences) is less than 1:25. Thus, a naive classification algorithm returning a negative answer for all the sentences will achieve 96.4% accuracy, but it is of no practical value. To evaluate the goodness-of-fit of our algorithm, we use the following measures:

- Precision = $\frac{TruePositive}{(TruePositive+FalsePositive)}$
- Recall = $\frac{TruePositive}{(TruePositive+FalseNegative)}$
- F-measure = $\frac{2 \times Precision \times Recall}{Precision + Recall}$

2.5 Parsing and feature engineering

2.5.1 Parsing

We use the Hebrew morphological parser developed by the National Institute for Testing and Evaluation (NITE) [2]. It is used to resolve morphological and parts of speech (POS) disambiguity. The reported accuracy of the NITE parser is 90% for the full morphological analysis and 95% for POS analysis.

Running the parser on the annotated student texts generates a tab-separated value file. Each row in the file corresponds to one word in the text, and contains the following information:

- isResearchQuestion: *True/False* - indicates whether the word is part of a research question sentence
- word original form: the word as appears in the text
- word basic form: the base form of the word
- POS: part of speech of the word

2.5.2 Bag of Words and feature set

First, we construct a Bag of Words (BOW) dictionary as follows:

1. Divide the data set randomly into training and test sets as presented in Table 1.
2. Build a BOW dictionary containing the basic form of each word that appears at least three times in a research question text segment (within the training set), and its corresponding POS.
3. Remove stop words: numbers, punctuation marks except for question mark, prepositions, pronouns, auxiliary verbs, all forms of the word "the" (could appear in a number of forms in Hebrew)

Then, for each sentence in the data set, we compute the following set of features:

- We introduce a feature for each BOW dictionary entry. The value of the features is defined as the number of appearances of the corresponding dictionary entry in the sentence.
- In addition, human experts composed a list of phrases that can be used as markers for a research question, such as "what is the connection", "what is the relation", etc. (in Hebrew, due to word agglutination, these phrases consist of two words only). We introduce an additional Boolean feature to represent the appearance of any of these phrases.

2.6 Results

We used the training set to train three types of classifiers: SVM, Logistic regression, and Naive Bayes. Their performance, computed over 500 5-fold cross-validation iterations, is presented in Table 2 (mean values). The best performance was achieved by the Logistic Regression classifier. To evaluate the performance on unseen data, we then trained a logistic regression classifier on the entire training set, and measured its performance on the test set. The results are presented in Table 3.

Table 2: The results of 500 5-fold cross validation runs on the training set

	Precision	Recall	F-measure
Logistic Regression	86.9%	74.3%	79.9%
SVM	75.9%	77.3%	75.9%
Naive Bayes	62.0%	88.6%	72.8%

Table 3: The results of the Logistic Regression model on the test set

	Precision	Recall	F-measure
Logistic Regression	84.2%	94.1%	88.9%

To understand the source of the errors we examined the sentences missed by the classifier. The main source of the errors is related to the failure of the parser to treat correctly a point sign '.' inside Latin names of organisms (e.g., 'E. Coli', 'St. Albus'). As a result, sentences containing such names were considered by mistake as two separate sentences and the classifier failed to identify them as a research question.

3. NEXT STEPS

Next, we plan to extend our model to identify the internal structure of the research question, as defined in the grading rubric. To support this step, the annotation scheme was extended to identify the required components of the research question: *opening* (e.g., "Our research question is:"), *independent variable* (e.g., ethanol concentration), *dependent variable* (e.g., cellular respiration rate), *connection between the variables*, and *organism* (e.g., bacteria, yeast). As this rubric is designed to be the basis for generating formative feedback, the experts gave a score (0-2) to each of these components, as well as an additional score for the *location* of the entire sentence in the text. This scheme was applied to 115 texts in a process similar to the one reported in Subsection 2.3. We also used the texts to create synthetic examples. In case the final version of a particular report was not well-written, the judges fixed the writing and inserted

the fixed version as an additional example. In total, 32 additional examples were created in this manner.

Based on this, we intend to create a computational model for identifying the internal structure, and use it to conduct an intervention study, in which students will receive formative feedback that is based on the computational analysis of the research question structure. In parallel, we will extend our method to the identify the remaining three discourse categories (biological process, research hypothesis, and description of the organism).

4. CONCLUSIONS

This paper presents preliminary results from a study that aims to develop NLP-based tools to assist teachers in providing formative feedback on scientific writing in Hebrew. Specifically, we demonstrate that our model can accurately identify the *research question* (or its absence), which is a key component of the specific writing task that we study (scientific report of inquiry project in Biology). Our results, although very preliminary, are a first step towards using NLP to provide formative assessment on scientific writing in Hebrew. To the best of our knowledge, there is no prior work that applies Hebrew NLP to provide formative feedback in K-12 science education.

5. ACKNOWLEDGMENTS

The authors thank Cipy Hofman and Yona Dolev for their contribution, and the National Institute for Testing and Evaluation (NITE) for providing access to the Hebrew morphological parser, and for partially funding this project. This research is supported by The Willner Family Leadership Institute for the Weizmann Institute of Science, Iancovici-Fallmann Memorial Fund, established by Ruth and Henry Yancovich, and by Ullmann Family Foundation.

6. REFERENCES

- [1] L. K. Allen, M. E. Jacovina, and D. S. McNamara. Computer-based writing instruction. In *Handbook of writing research*, pages 316–329. The Guilford Press, New York, 2016.
- [2] A. Ben-simon and Y. Cohen. The Hebrew Language Project : Automated Essay Scoring & Readability Analysis. In *IAEA Annual Conference*, January 2011.
- [3] J. Burstein, D. Marcu, and K. Knight. Finding the WRITE stuff: automatic identification of discourse structure in student essays. *IEEE Intelligent Systems*, 18(1):32–39, 2003.
- [4] Central Bureau of Statistics. Trends in Math and Science in Upper Secondary Education, 2006-2016 [Press release], 2018.
- [5] Y. Cohen, E. Levi, and A. Ben-Simon. Validating human and automated scoring of essays against “True” scores. *Applied Measurement in Education*, 31(3):241–250, 2018.
- [6] B. Galia Zer-Kavod Advisor and A. Yarden. *Thesis for the degree Doctor of Philosophy*. PhD thesis, Weizmann Institute of Science, 2017.
- [7] Israeli Ministry of Education. Syllabus of Biological Studies (10th-12th grade). Technical report, 2011.
- [8] A. Itai and S. Wintner. Language resources for Hebrew. *Language Resources and Evaluation*, 42(1):75–98, 2008.
- [9] G. J. Kelly and C. Bazerman. How students argue scientific claims: A rhetorical-semantic analysis. *Applied Linguistics*, 24(1):28–55, 2003.
- [10] H. McGarrell and J. Verbeem. Motivating revision of drafts through formative feedback. *ELT Journal*, 61(3):228–236, 2007.
- [11] National Research Council (NRC). A framework for K-12 science education: Practices, crosscutting concepts, and core ideas. Technical report, 2012.
- [12] R. H. Nehm, M. Ha, and E. Mayfield. Transforming biology assessment with machine learning: Automated scoring of written evolutionary explanations. *Journal of Science Education and Technology*, 21(1):183–196, 2012.
- [13] J. Osborne. Teaching scientific practices: Meeting the challenge of change. *Journal of Science Teacher Education*, 25(2):177–196, 2014.
- [14] N. Pendar and E. Cotos. Automatic identification of discourse moves in scientific article introductions. In *Proceedings of the Third ACL Workshop on Innovative Use of NLP for Building Educational Applications*, pages 62–70, 2008.
- [15] R. Porter, K. Guarienti, B. Brydon, J. Robb, A. Royston, H. Painter, A. Sutherland, C. Passmore, and M. H. Smith. Writing better lab reports. *The Science Teacher*, 77(1):43–48, 2010.
- [16] E. Riedel, S. L. Dexter, C. Scharber, and A. Doering. Experimental evidence on the effectiveness of automated essay scoring in teacher education cases. *Journal of Educational Computing Research*, 35(3):267–287, 2006.
- [17] R. D. Roscoe, L. K. Varner, S. A. Crossley, and D. S. McNamara. Developing pedagogically-guided algorithms for intelligent writing feedback. *Grantee Submission*, 8(4):362–381, 2013.
- [18] C. Strobl, E. Ailhaud, K. Benetos, A. Devitt, O. Kruse, A. Proske, and C. Rapp. Digital support for academic writing : A review of technologies and pedagogies. *Computers & Education*, 131:33–48, 2019.
- [19] C. Tansomboon, L. F. Gerard, J. M. Vitale, and M. C. Linn. Designing automated guidance to promote productive revision of science explanations. *International Journal of Artificial Intelligence in Education*, 27(4):729–757, 2017.
- [20] J. Wilson, R. Roscoe, and Y. Ahmed. Automated formative writing assessment using a levels of language framework. *Assessing Writing*, 34:16–36, 2017.
- [21] B. Woods, D. Adamson, S. Miel, and E. Mayfield. Formative essay feedback using predictive scoring models. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’17, pages 2071–2080, 2017.
- [22] M. Zhu, O. L. Liu, and H.-S. Lee. The effect of automated feedback on revision behavior and learning gains in formative assessment of scientific argument writing. *Computers Education*, 143:103668, 2020.
- [23] M. Zion, S. Cohen, and R. Amir. The spectrum of dynamic inquiry teaching practices. *Research in Science Education*, 37(4):423–447, 2007.

Discovering the Prerequisite Relationships Among Instructional Videos From Subtitles

Mehmet Cem Aytekin^{*}

Stefan Rübiger

Yücel Saygın

Sabancı University
Faculty of Engineering and Natural Sciences
Istanbul, Turkey

{mehmetaytekin, stefan, ysaygin}@sabanciuniv.edu

ABSTRACT

Nowadays, students prefer to complement their studies with online video materials. While there are many video e-learning resources available on the internet, video sharing platforms which provide these resources, such as YouTube, do not structure the presented materials in a prerequisite order. As a result, learners are not able to use the existing materials effectively since they do not know in which order they need to be studied. Our aim is to overcome this limitation of existing video sharing systems and improve the learning experience of their users by discovering prerequisite relationships among videos where basic materials are covered prior to more advanced ones. Experiments performed on commonly used gold standard datasets show the effectiveness of the proposed approach utilizing measures based on phrase similarity scores.

Keywords

prerequisite extraction, prerequisite graph, prerequisite

1. INTRODUCTION

With the widespread adoption of computers, especially among the young generation of students, and the video sharing platforms (VSP) such as YouTube, learners are more and more using video materials. In fact, there are many VSPs publishing learning material which are rich in content and very popular among students. The video lectures of the Physics Professor Walter Lewis¹ at MIT having millions of views in YouTube are an example of this paradigm shift.

Learning materials published on VSPs are not treated differently than other types of videos since these platforms are not designed to be used as an e-learning system. Therefore they do not present the materials in a structural manner following the prerequisite relationships. VSPs follow their users to

bring the most relevant personalized material, but these are not determined based on the background of their users, but just their interests. Therefore, the presented list of materials does not follow the prerequisite order. Our aim in this work is to overcome this limitation of existing VSPs by organizing the videos according to a prerequisite order, such that prerequisites are recommended to be watched prior to the actually searched material. This way we intend to improve the learning experience.

Our methodology is based on structuring the video learning materials using prerequisite relationships where basic materials are covered prior to more advanced ones. This is an offline process implemented as a separate module which can be integrated into any VSP providing an API with search capabilities. Given a predefined set of concepts, we first collect the video learning materials related to those concepts and extract their subtitles. We then build a model to infer prerequisite relationships based on the collection of subtitles. VSPs return a list of videos, where videos are ranked based on their relevance with respect to the search term. Our unsupervised methodology exploits the powerful relevance ranking models of the VSPs by incorporating the returned alternative materials in prerequisite relationship extraction. We implemented the proposed methodology using YouTube as a VSP. Experiments performed on concepts from a benchmark data set show that the proposed method utilizing measures based on similarity scores identifies the prerequisite relationships among those concepts and therefore provides users with a better learning experience.

2. RELATED WORK

Our related work is described in two main areas in the following subsections.

2.1 Prerequisite detection

The task of identifying prerequisite relationships between concept pairs was first introduced in [12] and existing methods that address this problem are based on supervised learning. One popular and important feature in this context is called reference difference (RefD) [3] which intuitively captures prerequisite relationships between concepts A and B by counting how often B refers to A and how often A refers to B . If B refers frequently to A , but A does not refer often to B , one may infer that B is a prerequisite for A . The original RefD feature relies on the hyperlink structure within documents, which is the reason for computing RefD based

^{*}Corresponding author

¹<https://www.youtube.com/watch?v=sJG-rXBmCc>

on Wikipedia articles. In addition to RefD, previous works [13, 8] extended the list of features derived from Wikipedia articles, e.g. by including related, but more abstract articles. In [6] word embeddings of texts are used as features besides 16 other features like RefD to represent text documents for prerequisite detection. Interestingly, RefD turned out to be consistently the most important feature across different languages and datasets, which motivates our choice for focusing on adapting RefD to unstructured video subtitles. In [1] a method is presented, which combines burst analysis and co-occurrence of words to identify prerequisite relationships. This approach uses unstructured text from books as input and it requires only light training as parameters need to be set based on the dataset, otherwise it relies on the default values. Unlike all previous methods, our method is fully unsupervised by nature. It relies on the core idea of RefD to determine prerequisite relationships, but in contrast to existing methods that exploit links in structured documents, we use exact matches to count how often concepts occur in unstructured text documents as noun phrases. Moreover, our approach could easily be integrated into the existing supervised methods as a feature.

2.2 Resources for extracting prerequisite relationships

In the past, different resources were used for identifying prerequisite relationships, namely text books [13, 4, 1], course prerequisites and video playlists [10], Wikipedia [12, 3, 5], a mixture of Wikipedia and video subtitles [8], and the Wikipedia clickstream [11]. Wikipedia has been the most popular resource as RefD relies on the structured information present in Wikipedia articles, e.g. links to related or more abstract concepts. But Wikipedia has multiple limitations as a resource. First, there might be no Wikipedia article for certain concepts [7]. Second, the desired concept might be part of a larger Wikipedia article which implies that some of the information is too broad or that concept simply cannot be found unless one knows the specific article in which that concept was mentioned. However, the most important limitation of Wikipedia in the context of e-learning is the fact that a concept is explained from a single perspective instead of multiple ones, which is important considering that individuals learn differently and might thus understand alternative explanations more easily. For these reasons, we opt in this paper for a VSP, YouTube in our case, as a resource for concepts since there are typically multiple videos available for a specific concept, potentially explaining it from different perspectives which benefits individuals as everyone learns differently. More precisely, we retrieve the subtitles of videos similar to [8], but in contrast to them, we collect a set of videos per concept instead of a single one per concept. Our approach is also different from [10], who utilize the downloaded video subtitles for creating bag-of-word representations to infer the hidden concepts using LDA and one video exists per concept.

3. MOTIVATION AND PROBLEM DEFINITION

As mentioned in Section 2.2, there may be no Wikipedia article available for a specific concept. Then any features including RefD relying on such structured text documents cannot be computed. For example, Wikipedia has no entry

for the concept "Recursive Backtracking" from our dataset (cf. Section 5.1), there is only an article related to the general concept of "Backtracking". Therefore, we extract the video subtitles and use them as text documents describing the concepts explained in the videos. Another advantage of using a VSP is that videos related to a concept explain the concept from different perspectives, with a varying level of detail. VSPs such as YouTube have powerful relevance ranking and diversification algorithms which we indirectly incorporate in the RefD score calculation by including the subtitles from the list of videos returned for a concept.

We model our problem with strictly partially ordered sets. Given a set of m concepts $C = \{c_1, \dots, c_m\}$ and a set of n videos associated with each concept, $V = \{v_{i,1}, \dots, v_{i,n}, \dots, v_{m,1}, \dots, v_{m,n}\}$, we extract from all collected videos related to a concept c_i , namely $\{v_{i,1}, \dots, v_{i,n}\}$, the subtitles and merge them into a text document t_i , such that each concept c_i is represented by a single text document t_i in the set $CT = \{(c_1, t_1), \dots, (c_m, t_m)\}$. From CT we form a strictly partially ordered set PO-CT by introducing the binary prerequisite relationship $Preq((c_i, t_i), (c_j, t_j))$ between c_i and c_j , where $c_i, c_j \in C$ and

$$Preq((c_i, t_i), (c_j, t_j)) = \begin{cases} 1 & \text{if } c_i \text{ is a prerequisite for } c_j \\ 0 & \text{otherwise} \end{cases}$$

Therefore, PO-CT is transitive (if c_i is a prerequisite for c_j and c_j is a prerequisite for c_k , c_i must also be a prerequisite for c_k), asymmetric (if c_i is a prerequisite for c_j , c_j cannot be a prerequisite for c_i), and irreflexive (c_i cannot be a prerequisite for itself) by definition [2]. Our final goal is to construct an acyclic prerequisite graph PG visualizing the prerequisite relations from PO-CT.

4. PREREQUISITE DISCOVERY PROCESS

Our method for building the prerequisite graph PG comprises two phases. In the first phase, we compute the strength of the pairwise prerequisite relationships which will be stored in a prerequisite matrix. Some of the relationships will violate the assumptions made for a partially ordered set, due to the pairwise computation of prerequisite relationships. For example, if $Preq((c_i, t_i), (c_j, t_j)) = 1$, $Preq((c_j, t_j), (c_k, t_k)) = 1$, $Preq((c_k, t_k), (c_i, t_i)) = 1$, then there would be a cycle of prerequisite dependencies as c_i would be a prerequisite for c_j , c_j would be a prerequisite for c_k , and c_k would be a prerequisite for c_i , which needs to be resolved. Therefore, in the second phase for graph construction, we use heuristics to overcome these issues.

4.1 Prerequisite Score Calculation

Determining if there is a prerequisite relationship between two concepts c_i and c_j implements the core idea of RefD, namely that if c_j occurs rarely in the text document t_i describing c_i , but c_i occurs frequently in the text document t_j representing c_j , then c_i is most likely a prerequisite for c_j . Unlike RefD, t_i and t_j do not contain related concepts to c_i and c_j , but rather describe only the concepts c_i and c_j . Since we compare text documents, we do not require any structured information such as links to related concepts. By gathering n number of videos for each of the concepts c_i and c_j from a VSP, our function $Preq()$ exhibits irreflexivity and asymmetry. We compute $Preq((c_i, t_i), (c_j, t_j))$ as follows:

1. Set input parameter - n : number of videos to collect per video for a concept
2. Given a pair of concepts c_i and c_j , retrieve the n most relevant videos for each of the concepts c_i and c_j from a VSP; extract their subtitles and merge those of $\{c_{i,1}, \dots, c_{i,n}\}$ into text document t_i and those of $\{c_{j,1}, \dots, c_{j,n}\}$ into t_j yielding (c_i, t_i) and (c_j, t_j) , respectively. t_i and t_j describe the concepts c_i and c_j in detail.
3. Preprocess t_i and t_j and create two lists L_i and L_j which contain all of the nouns and noun phrases from t_i and t_j , respectively. This step is performed since concepts occur in text documents always as nouns or noun phrases.
4. For each noun and noun phrase in L_i , count the exact matches with c_j and store it in a variable called $counts_j$.
5. For each noun and noun phrase in L_j , count the exact matches with c_i and store it in a variable called $counts_i$.
6. The output of the prerequisite relationship calculation is $w_{i,j} = counts_j - counts_i$
7. $RefD((c_i, t_i), (c_j, t_j)) = w_{i,j}$
8. Store $w_{i,j}$ in the score matrix W

The score matrix W has the following shape:

$$W = \begin{pmatrix} w_{1,1} & \cdots & w_{1,m} \\ \vdots & \vdots & \vdots \\ w_{m,1} & \cdots & w_{m,m} \end{pmatrix}$$

where $w_{i,j}$ corresponds to the prerequisite score between the concepts in the i -th row and the j -th column. Note that $w_{i,i}$, i.e. all elements on the diagonal, are zero due to the irreflexivity property of RefD. Moreover, $w_{i,j} = -w_{j,i}$ due to RefD being asymmetric. Due to this property, we have to compute $RefD((c_i, t_i), (c_j, t_j))$ only $m * (m - 1) / 2$ times. We also note that the output of RefD can be converted into a binary output as follows: If $w_{i,j} < 0$, c_i is a prerequisite for c_j and the strength of the prerequisite relationship is $|w_{i,j}|$. Otherwise c_j is not a prerequisite for c_i . In other words,

$$Preq((c_i, t_i), (c_j, t_j)) = \begin{cases} 1 & \text{if } w_{i,j} < 0 \\ 0 & \text{otherwise} \end{cases}$$

Therefore, $RefD((c_i, t_i), (c_j, t_j))$ approximates the binary relationship $Preq((c_i, t_i), (c_j, t_j))$.

4.2 Prerequisite graph construction

Given the score matrix W from Section 4.1, we want to construct the acyclic prerequisite graph PG where concepts correspond to nodes and directed edges from concept c_i to c_j with weight $w_{i,j}$ are added. However, since $RefD((c_i, t_i), (c_j, t_j))$ is a heuristic to approximate $Preq((c_i, t_i), (c_j, t_j))$, errors are introduced and PG constructed from W is not necessarily acyclic yet. For example, suppose that from the

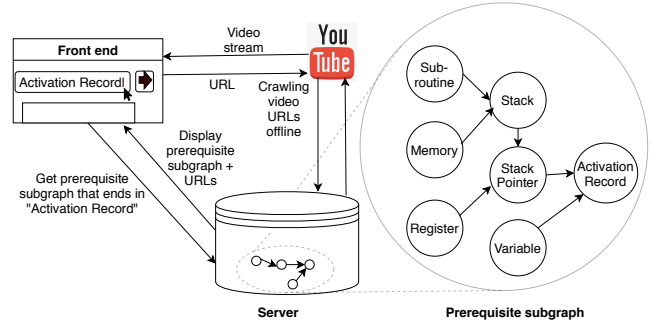


Figure 1: Client server architecture of our learning platform. Adapted from [9]

first phase, given three given concepts, a, b, c , we obtained the following matrix W :

$$W = \begin{pmatrix} 0 & x = -0.2 & -z = 0.2 \\ -x = 0.2 & 0 & y = -1.0 \\ z = -0.2 & -y = 1.0 & 0 \end{pmatrix} \quad (1)$$

The entries in W (cf. 1) correspond to the weights $x = w_{a,b}, y = w_{b,c}, z = w_{c,a}$, respectively. This matrix results in a PG with a cycle because a is a prerequisite for b (since $x < 0$), b is a prerequisite for c (since $y < 0$), and c is a prerequisite for a (since $z < 0$). To remove cycles, we apply to W the following method. Concept c_i , which is stored in the i -th row of W , is only connected to the prerequisite with the highest absolute weight $w_{i,j}^*$ in row i . If all weights are zero in row i , c_i has no outgoing edges. This way the most powerful prerequisite relationships are preserved.

This method only prevents cycle formation in the graph, but still allows to model scenarios like one concept being a prerequisite for multiple concepts or multiple concepts being prerequisites for a single concept. However, PG might still contain redundant edges after applying our method. For example, assume that we swap the weights of z in W (cf. 1), so $z = 0.2$ and $-z = -0.2$. Then our method results in a being a prerequisite for b and c , while b is a prerequisite for c . Now c is directly reachable from a , but also from a over b . To remove such redundant edges, we compute the transitive closure of the acyclic PG using Warshall's algorithm. The resulting PG can then be visualized.

4.3 Architecture and Implementation

We are in the process of integrating the methods described in Section 4 into our e-learning platform which uses YouTube videos as video learning materials. The platform is built on top of Open edX². In the context of the e-learning platform, the prerequisite relationships are extracted offline given a set of concepts, which allows us to construct the prerequisite graph PG from the score matrix W . A small sample PG is depicted on the right-hand side in Fig. 4.3 for the domain "Operating Systems". For example, to understand the concept "Activation Record", it is assumed that a learner knows about "Stack" and all the other concepts shown in the graph. Therefore, learners may only start "Activation Record" once they completed all prerequisites.

²<https://github.com/edx/edx-platform>

The rest of the client server architecture of our e-learning platform is depicted in Fig. 4.3. Initially, a set of concepts is automatically extracted from text documents such as books or slides according to [13]. URLs of video learning materials are then extracted from YouTube, together with the pairwise prerequisite relationships between the concepts based on the subtitles. Whenever a learner wants to study a concept, she submits a query through the front end, e.g. "Activation Record", and the query is then transferred to the server for processing. The server queries PG to return the subgraph which contains the requested concept and its prerequisites as a list of JSON objects, where each concept contains additional metadata like URLs to multiple YouTube videos and which of those should be recommended to be watched first by the learner, i.e. their rankings.

5. EVALUATION

The resulting PG depends on the quality of the identified prerequisite relationships. Therefore, for experiments we analyze the performance of our approach described in Section 4.1 in terms of how well it identifies prerequisite relationships according to the first phase of our methodology.

5.1 Datasets

For the experiments we used Metacademy³, which provides concepts for particular domains together with the prerequisite relationships among these concepts. Prerequisite relationships were annotated manually by experts of Metacademy. We focus on the domain "Data Structures & Algorithms" in our experiments which is comprised of 30 concepts from which we replaced three of them by three alternative ones that were listed as prerequisites for some of the concepts, but not included in the dataset. The main reason for this decision is due to them covering aspects of topics that are already included. From these 30 concepts, we randomly select 43 positive prerequisite relationship pairs for our experiments. In line with previous approaches [6, 8], we evaluate our method on a balanced dataset. Thus, we also generate 43 negative pairs by combining concepts that have no prerequisites in common. For each of the 30 concepts we retrieved the first n videos from YouTube and merged them into a single text document per concept, where $n = 1, \dots, 20$.

5.2 Performance for Prerequisite Detection

Our baseline method extracts the subtitles from a single video, whereas all other methods rely on merging the subtitles of multiple videos for a concept. We analyze how precision, recall, and F1-score of our proposed method are affected by varying n , the number of considered videos per concept c_i from which the subtitles are extracted to form the corresponding text document t_i .

The results are shown in Fig. 5.2. In terms of F1-scores, we observe that they gradually increase from 0.46, when using only subtitles of a single video per concept, up to 0.75 when incorporating subtitles from up to 20 related videos for a concept. Especially in the beginning, when using less than six videos per concept for subtitle extraction, adding more videos improves the F1-scores noticeable. But how does varying n affect precision and recall? Depending on

³<https://metacademy.org/browse>

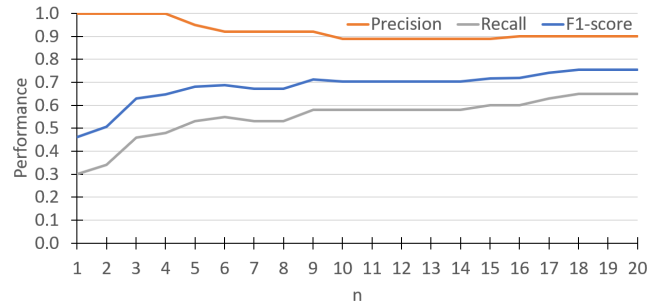


Figure 2: Influence of n , the number of considered videos per concept to be used for extracting subtitles, on the performance of our method.

the application, one of the two metrics might be more important. Fig. 5.2 indicates that precision slightly declines from 1.0 to 0.9 when considering more than 10 videos before stabilizing. However, at the same time recall roughly doubles from 0.3 to 0.65 when considering the 20 most relevant videos compared to using only a single video. Overall, the experiment suggests that including multiple videos per concept yields a more accurate detection of prerequisite relationships compared to using a single video per concept. One possible explanation for this increase in recall is that by including a larger number of videos, we also include a richer vocabulary as different educators prefer different terms. This, in turn, benefits the exact matches used in our method for detecting prerequisite relationships. One might even argue that this roughly corresponds to the idea of querying related Wikipedia articles instead of limiting one's computations to the Wikipedia articles describing the respective concept. However, this observation from our experiments might be an artifact and not hold for other domains and thus we cannot rely on this effect.

6. CONCLUSION

In this paper we have demonstrated that we can detect prerequisite relationships among video learning materials based on their subtitles using an unsupervised approach by utilizing the core idea of the well-known RefD metric with exact matches of concepts in subtitles that were collected from videos. Using only this indicator alone to determine prerequisites shows its effectiveness. This implies that our method could also be incorporated as a feature into supervised approaches to improve their performance.

One limitation of our proposed method is that it relies on exact matches and therefore ignores synonyms and semantically related terms that describe similar concepts. Therefore, it seems promising to support fuzzy matches in our method. One idea would be to employ word embeddings to that end in a similar fashion as described in [8]. Moreover, we have evaluated our proposed method only on a single domain thus far, but we plan to assess the performance on additional datasets from different domains. We hope our methodology of identifying the prerequisite relationship among video learning materials and presenting their related materials accordingly will improve the learning experience of students.

7. REFERENCES

- [1] G. Adorni, C. Alzetta, F. Koceva, S. Passalacqua, and I. Torre. Towards the identification of propaedeutic relations in textbooks. In *International Conference on Artificial Intelligence in Education*, pages 1–13. Springer, 2019.
- [2] C. Djeraba. *Mathematical Tools For Data Mining: Set Theory, Partial Orders, Combinatorics. Advanced Information and Knowledge Processing*. Springer, 2008.
- [3] C. Liang, Z. Wu, W. Huang, and C. L. Giles. Measuring prerequisite relations among concepts. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1668–1674, 2015.
- [4] C. Liang, J. Ye, S. Wang, B. Pursel, and C. L. Giles. Investigating active learning for concept prerequisite learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [5] C. Liang, J. Ye, H. Zhao, B. Pursel, and C. L. Giles. Active learning of strict partial orders: A case study on concept prerequisite relations. In M. C. Desmarais, C. F. Lynch, A. Merceron, and R. Nkambou, editors, *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada, July 2-5, 2019*. International Educational Data Mining Society (IEDMS), 2019.
- [6] A. Miaschi, C. Alzetta, F. A. Cardillo, and F. Dell’Orletta. Linguistically-driven strategy for concept prerequisites learning on italian. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 285–295, 2019.
- [7] C. Okoli, M. Mehdi, M. Mesgari, F. Å. Nielsen, and A. Lanamäki. Wikipedia in the eyes of its beholders: A systematic review of scholarly research on wikipedia readers and readership. *Journal of the Association for Information Science and Technology*, 65(12):2381–2403, 2014.
- [8] L. Pan, C. Li, J. Li, and J. Tang. Prerequisite relation learning for concepts in moocs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1447–1456, 2017.
- [9] S. Rübiger, T. Dalkılıç, A. Doğan, B. Karakaş, B. Türetken, and Y. Saygı. Exploration of video e-learning content with smartphones. *International Association for Development of the Information Society*, 2020.
- [10] S. Roy, M. Madhyastha, S. Lawrence, and V. Rajan. Inferring concept prerequisite relations from online educational resources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9589–9594, 2019.
- [11] M. Sayyadiharikandeh, J. Gordon, J.-L. Ambite, and K. Lerman. Finding prerequisite relations using the wikipedia clickstream. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 1240–1247, 2019.
- [12] P. P. Talukdar and W. W. Cohen. Crowdsourced comprehension: predicting prerequisite structure in wikipedia. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 307–315. Association for Computational Linguistics, 2012.
- [13] S. Wang, A. Ororbia, Z. Wu, K. Williams, C. Liang, B. Pursel, and C. L. Giles. Using prerequisites to extract concept maps from textbooks. In *Proceedings of the 25th acm international on conference on information and knowledge management*, pages 317–326, 2016.

A method for generating features representing the students' degree of anticipation/delay to complete assignments

Francisco Cervera-Mercadillo,
Juan A. Lara
Madrid Open University, UDIMA
panchitopin@gmail.com,
juanalfonso.lara@udima.es

ABSTRACT

Some current methodologies stress the importance of continuously assessing the students to check their progress, instead of having an only final examination. To do so, several learning resources are presented to the students, so they can complete different actions over those resources at certain particular dates. This scenario presents a good chance for research since it might be useful to analyze the point up to which students complete in time the tasks they are supposed to do. In this paper, we present a method that takes raw log Moodle data and generates new features that represent the degree of anticipation/delay of students when completing the tasks suggested by their tutor. We have developed a system that implements this method obtaining some minable sights that preliminarily seem to be useful to predict phenomena such as academic dropout. Obviously, future deep experiments must be conducted to demonstrate the validity of those new features.

Keywords

Data preprocessing; E-learning; Continuous Assessment; Assignment-related temporal features; Moodle.

1. INTRODUCTION

In the last years, new formative paradigms such as E-learning (Electronic Learning) have emerged in order to provide people with ubiquitous learning [1]. E-learning platforms like Moodle provide useful data about students' behaviors that can be exploited by Educational Data Mining (EDM) or Learning Analytics (LA) techniques [2].

Many current education initiatives are based on continuous assessment during the courses. It means that students are encouraged to complete different assignments at certain suggested dates [3]. Some of those students will complete those assignments at the suggested date, others will do it earlier and some of them may complete it late.

In this paper, we present a method that takes Moodle logs from a particular course and a list of suggested dates where assignments are suggested to be completed by the students according to the

tutor recommendation and generates a minable sight in the form of a table containing as many rows as students enrolled in the course and as many columns as assignments or tasks that students should carry out. Each cell will take an integer value representing the degree of anticipation or delay for the particular student (row) to complete each task (column) suggested by the tutor. These new features can be useful in the prediction of the students' performance, which is one of the main goals of EDM.

Although there is some literature on procrastination [4], to the best of our knowledge, it is the first time that the high-level features we propose have been used in EDM field so our ideas may represent an interesting line of research, which is the main contribution of this paper.

The rest of this paper is organized as follows. Section 2 presents a brief description of the proposed method. Section 3 contains a technical description of the implemented tool and the preliminary results obtained. Finally, Section 4 includes a discussion of the results and conclusions obtained, as well as some potential future lines of research.

2. METHOD

In this paper, we propose a method that intends to generate new features from Moodle logs that represent the degree of anticipation or delay for each student to complete the tasks proposed by the tutor in order to reach the formative results supposed to acquire during continuous evaluation.

Those suggestions may be really diverse and include tasks such as reading a document, watching a video, taking part in a forum or submit a report. In particular, we have defined a series of potential kinds of tasks that the students can carry out with educational resources uploaded to the Moodle virtual classroom by the tutor. Those are: View, Create, Update, Delete, Subscribe, Review, Submit, and Start. For each educational resource and type of task, the tutor has to define a reference date when the students are supposed to complete that task on the respective resource (Table 1).

The proposed method intends to take raw Moodle data and generate a new feature for each resource and type of task in a way that each new attribute will take negative values for "eager" students, positive values for "late" students, and 0 values for students who complete their tasks exactly the same day as the tutor suggested.

This method is intended for EDM/LA experts who intend to use these kinds of features in order to analyze their impact on students' performance. Note that our method starts from Moodle

Francisco Cervera and Juan Lara "A method for generating features representing the students' degree of anticipation/delay to complete assignments" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 574 - 577

log data that register all the events (actions) carried out by the students on each resource of the Moodle platform.

Table 1. Excerpt of an example of suggested tasks and dates

Computer Architecture Course	
<i>Week 1</i>	
Task	Suggested Date
View the Presentation Session (VPS)	8-Oct-2019
Read the Teaching Guide (RTG)	13-Oct-2019
<i>Week 2</i>	
Task	Suggested Date
Subscribe to Doubts Forum (SDF)	14-Oct-2019
Visualize Unit 1 PDF Document (VU1)	16-Oct-2019
Submit the report of Activity 1 (SA1)	20-Oct-2019
...	...

We have also defined our approach in a way that the expert can perform the following tasks to particularly design his/her data analysis:

1. Student selection (focus only on some certain students).
2. Period selection (focus only on certain time intervals of the course).
3. Resource selection (only consider some educational resources).
4. Task type selection (only consider some types of actions for each resource).
5. Reference dates definition (define, for each selected resource and type of action).
6. Generate new attributes' values (for each selected resource, type of action and student).
7. Manually add the class attribute for the latter analysis (dropout, pass/fail, ...).

That would lead to a minable sight for predictive purposes with the newly generated features. Table 2 shows an example.

Table 2. Excerpt of an example of generated minable sight for academic dropout prediction

	New Generated Features						Dropout
	Week 1		Week 2			...	
Student	VPS	RTG	SDF	VU1	SA1	...	
<i>Student₁</i>	0	-2	0	-1	-3	...	0
<i>Student₂</i>	+2	+3	0	+8	+12	...	1
...
<i>Student_n</i>	+1	0	+2	-1	0	...	0

3. IMPLEMENTED SYSTEM AND RESULTS

In this section we will explain the main aspects about the design of the system (3.1), how the system works (3.2) and system outputs (3.3).

3.1 Design and Technical details

The system was designed in order to meet four important main objectives:

1. The implemented method should guide the user step by step, executing the different tasks developed as an assistant, to finally obtain the desired minable view.
2. Due to the problem of dealing with large volumes of records to be processed, data persistence was decided not to be necessary. In our case, preprocessing the data in memory helped speed up the application of the different data selection, cleaning and transformation techniques.
3. During the development of the application, it was intended to design a system focused on usability, in order to facilitate the learning and use of the tool by the end-user.
4. It was necessary to define a properties file so the user could define some parameters difficult to provide a value for in execution time.

To achieve these objectives, the implementation was based on a web development paradigm with generally visual components (selects, multi-selects, calendars for date selection), except those values that are necessary to enter into the system for the creation of new high-level features, which inevitably force the introduction of the necessary values.

From a technical point of view, the implementation of the tool was carried out under the Java development language (JDK 1.8), mainly due to the multiple capabilities and features it offers. Since our system is a web application, we relied on its main framework, Spring 5.0, granting an agile development, based on the injection of dependencies and therefore, decoupled and easily scalable. This framework offers by default the Thymeleaf 3.0.4 template engine for the creation of the different views and, in our opinion, perfectly meets the needs imposed, as it allows the definition of reusable fragments and layouts, as well as a wide set of expressions to directly deal with the different data models generated at each step or task performed.

We also used the Bootstrap 4.0.0 framework, which has facilitated the design of the application interfaces, providing the web application with responsive features necessary to adapt the tool to the different existing viewing platforms. Apache Maven 4.0.0 let us manage and build the project in a simple way, as well as define the necessary dependencies of the system, obtaining them directly from its central repository.

Other technologies or frameworks such as jQuery (improvement of the interactivity of the application with the end-user), JUnit 5, Mockito (unit tests of validation of the most important methods of the application and integration of components) and Docker (creation of lightweight containers and highly portable for deployment) were used for the development of the project.

3.2 System working

As already mentioned, the system defines a series of tasks to be performed as an assistant. The system will guide the expert through the following implemented tasks, which are listed and detailed below, in order to be able to design a personalized educational data analysis:

1. Loading of the Moodle log file (Comma Separated Values -CSV- file) of a specific subject, that is, the source dataset to be preprocessed.
2. The expert selects those students not to be included in the final dataset.
3. The system allows the expert to select a date range, discarding all those records that are not inside that period. The system will not allow selecting dates outside the dates range existing in the dataset.
4. The expert selects those resources that he or she does not want to include in his analysis.
5. The system allows the grouping of resources, in order to consider them, from that moment, as the same resource.
6. The system shows for each resource, the different types of actions existing in the dataset, and the expert selects those that he or she wishes to remain in the final dataset (Figure 1 shows a screenshot of this step).

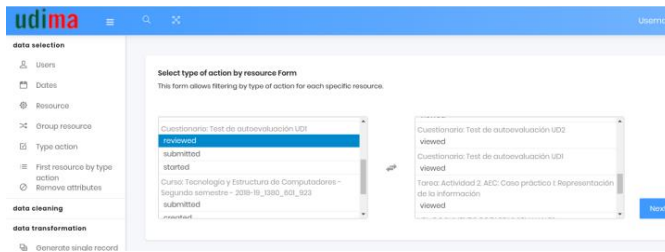


Figure 1. Selection of types of actions

7. The expert establishes the reference date for each resource and type of action (see Figure 2).

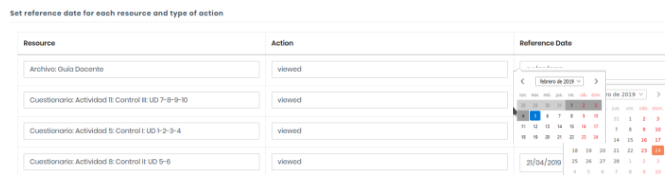


Figure 2. Reference dates definition

8. The system selects the first existing action type for each resource and user.
9. The system generates for each student a single record where, for each resource and type of action, it creates an entire attribute. This attribute represents the difference in days, between the reference date and the date recorded in the input file (see Figure 3, where students' names have been anonymized).

User	Sesion1	Read U1	Activity1
Student 1	1	65	47
Student 2	-2	7	14
Student 3	-6	-2	-4
...	3	63	42

Figure 3. Attribute values system generation

10. The system allows the expert to give value to some class attributes for classification purposes (see Figure 4).

User	Final_Mark	Examination_Taken	Numerical_rating	Nominal_rating	Dropout
Student 1	Pass	Yes	8	Notable	No
Student 2	Pass	Yes	9	Sobresaliente	No
Student 3	Fail	Yes	4	Suspense	No
...	Fail	No	0	Suspense	Yes

Figure 4. Class attributes user value definition

11. The expert exports the results obtained to a file with CSV format.

3.3 System output

The system generates a CSV file containing one record per user (anonymized), indicating the degree of advancement or delay in the performance of the activities selected by the expert, according to the date proposed by the tutor, as well as the new features created for classification. Figure 5 shows an example.

```
"User","Teaching Guide viewed","Activity11 viewed","Activity5 viewed","Activity8 viewed",-
"0","-3","1","65","47","3","-1","4","14","1","Pass","Yes","8.0","Remarkable","No"
"1","-2","-2","7","14","-1","11","-5","-2","-3","Pass","Yes","9.0","Outstanding","No"
"2","-3","-6","-2","-4","-2","2","-5","-3","Fail","Yes","4.0","Pass","No"
"3","-27","3","63","42","-1","2","21","8","90","Fail","No","0.0","Fail","Yes"
```

Figure 5. Example of CSV file generated by the system

After conducting some preliminary experiments, it was possible to obtain some interesting minable sights and get some predictive models using Weka¹. Applying, for example, a classification algorithm based on decision trees, we obtain an interesting model to predict whether a student will pass or fail the course. Below, we show the results obtained for this example case:

```
Decision Stump-Classification
Case Study I_viewed <= -2.0 : Pass
Case Study I_viewed > -2.0 : Fail
```

If we look at the predictive model returned by Weka, we find that the activity "Case Study I" becomes especially relevant among all the resources and types of actions of the original dataset. It indicates that all those students who visualize this activity with an advance of two or more days, with respect to the reference date established by the tutor, will finally pass the subject. Otherwise, they will fail.

¹ <https://www.cs.waikato.ac.nz/ml/weka/>

This acquired knowledge could directly influence the decisions that teachers should make regarding the activity, monitoring their development through tutorials with students, reinforcing the teaching material, encouraging students to carry it out or other types of actions.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a method that allows generating new features that represent the degree of anticipation or delay of students when completing the tasks suggested by the tutor. Our method is applicable in open courses that adopt methodologies based on the existence of a tutor that guides students by defining a series of continuous tasks that students are encouraged to complete at a certain date.

We have developed a preliminary system that implements that method with Moodle data, and also gives the user the possibility of performing some preprocessing tasks, such as student selection, resource selection, and so on. We have conducted preliminary experiments to obtain some minable sights of different high education open courses. Those tests make us be optimistic about the usefulness of the newly generated attributes and their potential application for future research.

The main future line of research that we should carry out next is the application of the method in different courses and analyze if the new proposed features are valid to predict important educational phenomena such as students' dropout, students' final marks, and so on.

Once a stable release of the system is finished and tested, we intend to provide the community with an access URL so that the system can be publicly used and tested.

5. REFERENCES

- [1] Cárdenas-Robledo, L. A. and Peña-Ayala, A. 2018. Ubiquitous learning: A systematic review. *J. Telemat. Inform.* 35, 5, 1097-1132. DOI=<https://doi.org/10.1016/j.tele.2018.01.009>.
- [2] Romero, C. and Ventura, S. 2020. Educational data mining and learning analytics: An updated survey. *WIREs Data Mining Knowl Discov.* e1355. DOI=<https://doi.org/10.1002/widm.1355>
- [3] Burgos, C., Campanario, M. L., de la Peña, D., Lara, J. A., Lizcano, D. and Martínez, M. A. 2018. Data mining for modeling students' performance: A tutoring action plan to prevent academic dropout. *Comput. Electr. Eng.* 66, 541-556. DOI=<https://doi.org/10.1016/j.compeleceng.2017.03.005>.
- [4] Hooshyar, D., Pedaste, M. and Yang, Y. 2020. Mining Educational Data to Predict Students' Performance through Procrastination Behavior. *Entropy* 22, 12. DOI=<https://doi.org/10.3390/e22010012>

Predicting students' performance using emotion detection from face-recording video when interacting with an ITS

Wilson Chango¹, Miguel Sanchez-Santillan², Rebeca Cerezo², Cristóbal Romero³

¹Pontifical Catholic University of Ecuador, Ecuador

²University of Oviedo, España

³University of Cordoba, España

wilson.chango@pucese.edu.ec, sanchezsmiguel@uniovi.es, cerezorebeca@uniovi.es, cromero@uco.es

ABSTRACT

This research aims to predict the academic performance of students when interacting with an Intelligent Tutoring System (ITS) from emotions detection and analysis. We use data from 47 university students in a virtual learning environment. We have used data gathered from face recording of students' interactions with the system to detect students' emotions and determine to what extent they can predict the final students' performance during the learning session.

Keywords

Predicting performance, Emotion detection, Video analytics.

1. INTRODUCTION

Emotions are a critical component of learning and problem solving, especially when it comes to interacting with computer-based learning environments (CBLEs) [5]. Studies from affective computing literature suggest that facial expressions may be the best single method for accurately identifying emotional states [4]. The automatic detection of emotions techniques are capable of isolating the mood of a learner by means of a facial recognition system through artificial intelligence and there are already tools that enable the processing of data in the form of video, such as the Microsoft Emotion API [1], FaceReader™, etc. However, we have not noticed previous studies testing to what extent the emotion recognition result of these tools is powerful enough to predict student's performance. It could be potentially contributing to enhance the quality and efficacy of CBLEs (e-learning, multi-agent systems, intelligent tutoring systems, serious games, etc.) by including the learner's emotional states.

This research aims to test if student's emotions recognized by and API during a learning session with an ITS can be enough to predict the final student's performance.

2. EXPERIMENTS

Data were collected from 47 undergraduates enrolled at a public university in the north of Spain whom learned about a complex science topic while interacting with the ITS MetaTutorES [3] a

multi-agent computerized learning environment. Participants represented a variety of disciplines, including psychology, education and engineering. The emotion data collected was naturally occurring, the emotions arose from interactions with a the ITS MetaTutorES, designed to teach learners about the human circulatory system during a session ranging from 2:30 to 3:00 hours. During and at the end of the session, performance test about the circulatory system knowledge were taken for every subject, giving a final performance value ranging between 0 and 10, showing 10 the best performance. A pretest about previous circulatory system knowledge is taken at the beginning of the session and final performance is corrected based on that previous level. Videos from every learner's facial expressions were captured with a webcam and analyzed using automatic facial recognition software (Microsoft Emotion API [1]). The API classifies the facial expression in eight classes of emotion: anger, contempt, disgust, fear, happiness, neutral, sadness, and surprise. The analysis allows us to obtain at least one highest emotion during the learning session from every students' frame having a high volume of frames (1 frame for second) for each student in every session. The confidence (values between 0 and 1) gives the likelihood for each class of emotion.

The first step of the experiment consists on check the correlations between the emotions detected and the students' performance. The Pearson correlation test examines the relationship of each emotion with the student's performance obtained during the learning session. The R value in Pearson's correlation coefficient goes from -1 to 1, meaning both values a high level of correlation and 0 a null level of correlation between variables (See Table 1).

Table 1: Pearson correlation test results

Emotion	R-Value
Anger	0.1295
Contempt	0.2165
Disgust	0.0882
Fear	-0.2415
Happiness	0.0459
Neutral	0.0463
Sadness	0.1546
Surprise	-0.1062

According to the results of table 1 none of the variables is highly correlated with the performance. However, based on the axes of emotions valence -positive emotions (happiness); negative emotions (anger, contempt, fear, disgust); non valence (neutral and surprise) [6] and looking at the positive or negative relationship, we can observe that only negative or non valence emotions are negatively related with performance.

Wilson Gustavo Chango Sailema, Miguel Sánchez, Rebeca Cerezo and Cristóbal Romero "Predicting students' performance using emotion detection from face-recording video when interacting with an ITS" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 578 - 580

In the second step, we applied several classification algorithms using the 8 emotions as input attributes for predicting the student's final performance. We used white box classification models (decision trees and rule induction algorithms) because the models they produce (tree and IF-THEN rules) are easy to understand [7]. In our experiments, we selected six well-known classification algorithms provided by WEKA [8]: three decision tree algorithms and three rule induction algorithms (see Table 3).

Table 3: Decision Trees classification algorithms.

Type	Algorithms	Description
Trees	J48	Java implementation of C4.5.
	Reptree	Fast tree learner that uses reduced-error pruning.
	Randomtree	Construct a tree that considers a given number of random features at each node.
Rules	Jrip	RIPPER algorithm for fast, effective rule induction.
	Nnge	Nearest-neighbor method of generating rules using generalized exemplars.
	Part	Obtain rules from partial decision trees built using J4.8

We executed each algorithm using stratified 10-fold cross-validation in which the dataset is randomly divided into 10 disjointed subsets of equal size in a stratified manner. We have compared the test results using the Accuracy and ROC Area evaluation measures (see Table 4).

Table 4. Results produced by all algorithms.

Algorithm	% Accuracy	ROC Area
Jrip	63,8298	0,5820
Nnge	53,1915	0,5290
Part	63,8298	0,6590
J48	63,8298	0,6770
Reptree	48,9362	0,5170
Randomtree	59,5745	0,5950
Avg	58,8653	Error de sintaxis, 0,5932

Table 4 shows that the best results (highest values) were produced by J48 (63,8298%Acc and 0.6770 AUC). Next, we show in Table 5 the obtained decision model by J48 algorithm.

Table 5. J48 decision tree.

Contempt <= 0.126904: Pass
Contempt > 0.126904
Disgust > 0.137741
Sadness <= 0.1977232
Fear <= 0.1551857: Pass
Fear > 0.1551857: Fail
Sadness > 0.1977232: Fail
Disgust <= 0.137741: Pass
Number of Leaves : 5
Size of the tree : 9

The Table 5 show us a decision tree that let us learn some interesting information from. On one side, students who Pass show lower values than an umbral of emotions contempt, disgust, fear and disgust, and students who Fail show higher values than a umbral of these emotions.. On the other side, we can observe that negative emotions have more prediction power on performance than positive or non-valence emotions. And finally, negative emotions values over 0.15 (15% of the session time) are defintory to a Fail ending.

3. CONCLUSIONS

There was an assumption that emotions experienced during complex learning will impact learning and problem solving [2], and therefore, achievement. However, in this study, we observe that student's emotions when interacting with an ITS are not enough for predicting students' final performance. The results give us some information the relationship of each emotion with the student's performance. However could be necessary to refine and redefine the API emotions classification based on an educational psychology framework for some close emotions (e.g attention, engagement, hope, pride, etc.).

Finally, we purpose as a future prospect adding other different variables/attributes from the interaction with the ITS such as log files, eye tracking, etc. in order to obtain higher accuracy values to predict students' performance. We also want to use more classifiers algorithms, particularly deep learning which would perform significantly better than classic methods.

4. ACKNOWLEDGMENTS

The authors acknowledge the financial subsidy provided by Spanish Ministry of Science and Technology TIN2017-83445-P and Plan de Ciencia Tecnología e Innovación del Gobierno del Principado de Asturias (FC-GRUPIN-IDI/2018/000199).

5. REFERENCES

- [1] Arora, R. et al. 2018. Microsoft Cognitive Services. *International Journal of Engineering Science and Computing*. 8, No.4 (2018), 17323–17326..
- [2] Buder, J. and Hesse, F.W. 2017. *Informational environments: Effects of use, effective designs*. Springer.
- [3] Cerezo, R. 2018. MetaTutorES: Evaluación e intervención en metacognición desde una perspectiva multimodal. Symposium conducted at the *IX International Congress of Psychology and Education* (Logroño, Spain, Jun. 2018).
- [4] D'Mello, S. and Kory, J. 2012. Consistent but Modest: A Meta-Analysis on Unimodal and Multimodal Affect Detection Accuracies from 30 Studies. *Proceedings of the 14th ACM International Conference on Multimodal Interaction* (New York, NY, USA, 2012), 31–38.
- [5] Harley, J.M. et al. 2015. A multi-componential analysis of emotions during complex learning with an intelligent multi-agent system. *Computers in Human Behavior*. 48, (Jul. 2015), 615–625.
- [6] Pekrun, R. 2011. Emotions as drivers of learning and cognitive development. *New perspectives on affect and learning technologies*. Springer. 23–39.
- [7] Romero, C. et al. 2013. Web usage mining for predicting final marks of students that use Moodle courses. *Computer Applications in Engineering Education*. 21, 1 (2013), 135–146. DOI:<https://doi.org/10.1002/cae.20456>.

- [8] Witten, I.H. et al. 2011. *Data Mining: Practical Machine Learning Tools and Techniques second edition.*

Applying Recent Innovations from NLP to MOOC Student Course Trajectory Modeling

Clarence Chen
University of California, Berkeley
clarencechenct@berkeley.edu

Zachary Pardos
UC Berkeley School of Information
zp@ischool.berkeley.edu

ABSTRACT

This paper presents several strategies that can improve neural network-based predictive methods for MOOC student course trajectory modeling, applying multiple ideas previously applied to tackle NLP (Natural Language Processing) tasks. In particular, this paper investigates LSTM networks enhanced with two forms of regularization, along with the more recently introduced Transformer architecture.

Keywords

next-step prediction, predictive modeling, course trajectory, mooc, lstm, transformer

1. INTRODUCTION AND MODEL OUTLINES

1.1 Fundamentals of Predictive Modeling

Recent innovations in deep learning methods for NLP (Natural Language Processing) tasks such as [7] in the past few years have consistently pushed the state of the art in a wide range of benchmark NLP tasks, while yielding new strategies that can be applied to predictive modeling tasks in a more general sense. This is because the majority of these NLP tasks within the scope of these innovations can be parameterized in terms of modeling the function f in the equation

$$P(y|x_0, \dots, x_t) = f(x_0, \dots, x_t; \theta) \quad (1)$$

where f is a probability mass function with parameters θ over the random variable y , and x_0, \dots, x_t , drawn from a discrete set of tokens T , represent the context from previous time steps. Unfortunately, there is little literature in the domain of education analytics exploring the effectiveness of innovations from NLP for education analytics tasks that also conform to this predictive modeling paradigm. Nevertheless, the LSTM (Long-Short-Term Memory) DNN (Deep Neural Network) architecture, an earlier innovation which was the architecture of choice for predictive modeling tasks in NLP before the past few years, has been successfully applied to several education analytics tasks. These papers demonstrate potential for further exploitation of the similarities between

predictive modeling tasks in education analytics and NLP, while providing a baseline to compare with more recent innovations presented in this paper.

1.2 Previous Work with MOOC Course Trajectory Modeling

One of the first papers to present an application of DNN models for predictive modeling tasks in education analytics is [4], where the authors specifically investigate the applicability of DNN models for modeling student course trajectories in MOOCs (Massive Open Online Courses). Specifically, the authors of [4] demonstrate the effectiveness of LSTM DNN models for this task over other strategies, such as using n -gram models that condition their predictions over small number of past course nodes. Finally, the authors provides suggestion for incorporating such a predictive model in a wider context, including tie-ins with the MOOC service to provide user-facing suggestions and live feedback to monitor the predictive model's performance.

1.3 Baseline LSTM

This model is identical to the Baseline LSTM model featured in [4], using the same LSTM architecture and a nearly identical hyperparameter set and training scheme, further detailed in Section 2.2. This model is intended as a control baseline to assess the performance of other models tested in this paper.

1.4 Transformer Architecture

As noted in Section 1, the shared abstraction of both course trajectory modeling and many NLP tasks as a discrete next-step predictive task suggests applying innovations from NLP to improve performance in course trajectory modeling. In particular, the Transformer architecture, first featured in [7], is one such major architectural innovation.

Transformer Architecture Details. In [7], the authors construct a DNN model architecture centered around modular Transformer blocks as described in Section 1.4. In contrast with the need for $O(n)$ forward and backward passes per input sequence through each of the LSTM recurrent nodes, the entire Transformer model is designed to only require one forward and backward pass through the entire model to process each input sequence. In addition to forming next-step predictive models from these transformer blocks, The authors also provide additional architectural topologies for

Clarence Chen and Zachary Pardos "Applying Recent Innovations from NLP to MOOC Student Course Trajectory Modeling" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 581 - 585

models tailored to other tasks such as machine translation or text classification, emphasizing the applicability of the Transformer blocks for a diverse array of NLP tasks. Please refer to [7] for more information about the composition of a Transformer-based next step predictive model.

Multi-Head Dot-Product Self-Attention. The multi-head dot product self-attention mechanism is the core architectural innovation which enables a Transformer block to fit to temporal correlations present in the training set in one forward and backward pass. On an abstract level, the operation used by the dot product self-attention mechanism with h heads to compute temporal correlations is the the scaled and masked outer product of each $k_i, q_i : i \in \{1, \dots, h\}$ (derived from the input tensor $x \in \mathbb{R}^n \times \mathbb{R}^d$) as shown in Equation (2):

$$t_i = \text{softmax} \left(\frac{\text{mask}(k_i q_i^\top)}{\sqrt{[d/h]}} \right) \quad (2)$$

The results $t_i \in \mathbb{R}^n \times \mathbb{R}^n : i \in \{1, \dots, h\}$ then directly capture how the features of each k_i and q_i are correlated over each pair of time steps in the input sequence. Note that the operator $\text{mask}(\cdot)$ in Equation (2) zeros out lower triangular entries in $k_i q_i^\top \in \mathbb{R}^n \times \mathbb{R}^n$, corresponding to the dot product of features in q_i with features in k_i from previous time steps. Please refer to [7] for more information about the multi-head dot product self-attention mechanism and the the Transformer block as a whole.

1.5 LSTM Enhancements

This section features two different enhancements featured in Kirill Mavreshko's¹ implementation of a Transformer-based next-step predictive model that could be independently used with LSTM models to yield performance improvements for student course trajectory modeling in MOOCs. These enhancements have also been independently backed with theoretical justification and empirical experiments, demonstrating performance improvements in coordinated NLP tasks when applied to LSTM models, as further detailed in [2] and [5].

Confidence Penalty Term in Loss. The baseline LSTM model already features some form of regularization, particularly dropout in the weights of the recurrent layers during training. However, for any classification task with a correct output label y_{true} in a set of possible labels T , examining the equation for the cross-entropy classification loss

$$L(\theta) = -\log P(y_{true}; \theta) = -\log p_{true} \quad p_j = P(y_j; \theta) \quad \forall j \in T \quad (3)$$

suggests an additional regularization term that penalizes highly confident distributions to reduce overfitting. The confidence penalty uses $H(p(y; \theta))$ as quantitative measure of confidence in a model's output distribution, where a higher value represents a lower level of confidence that the model predicts for each outcome $j \in T$. As a result, the new loss

¹Copyright 2018 by Kirill Mavreshko. Source code at <https://github.com/kpot/keras-transformer>

function expands to

$$L^*(\theta) = -\log p_{true} - \beta H(p(y; \theta)) = -\log p_{true} + \beta \sum_{j \in T} p_j \log p_j \quad (4)$$

where β is a scalar hyperparameter weight for the Confidence Penalty loss term. For theoretical arguments and empirical evidence for adding a confidence penalty term, please refer to [5].

Tied Embedding Layers. Another opportunity to introduce additional regularization to any sort of discrete next-step predictive model is found when examining the model's embedding and output layers, specifically

- The embedding layer L with dimension $|T| \times d_{embed}$, mapping input tokens in T to vectors in a latent feature space.
- The output layer W with dimension $d_{final} \times |T|$, mapping the final intermediate layer output h to an probability logit over the set of all input tokens T .

After enforcing the condition $d_{embed} = d_{final}$ by inserting a feed-forward layer between the rest of the model and the output later, W is tied to the embedding layer L fixing $W = L^\top$. For theoretical arguments and empirical evidence for the effectiveness of tying the output layer in this fashion, please refer to [2].

2. EXPERIMENTS AND EMPIRICAL RESULTS

2.1 Dataset Cleaning and Processing

2.1.1 Procedures for Dataset Cleaning and Processing

Given that the task of student course trajectory modeling requires predicting where a student will navigate next given the student's previous navigation patterns, extensive processing of raw MOOC server logs is required before any training can occur. This process is explained in great detail in [4], with the main steps listed below:

1. Given the raw server log records, select the `basic_action` column, timestamp, username, and title columns necessary to build unique course node tokens in step 3.
2. Filter out all log records except those with `basic_action` label `seq_next`, `seq_prev`, or `seq_goto`, representing the full set of navigation actions a student can take for each of the MOOCs.
3. Construct a unique positive integer token ID for each course node through concatenating each component of the full course path to construct a unique name for each course node, then assigning each unique name to the token ID.
4. Assemble the full sequence of navigation records for each user by grouping by user ID, then ordering within in each group by timestamp.

Table 1: MOOC Course Trajectory Dataset Summary Statistics

Institution	Course	Term	Nodes	Users
DelftX	AE1110X	Fa. 2015	291	14496
UCBX	EE40LX	Fa. 2015	287	30633
UCBX	Fin101X	Sp. 2016	114	2951
UCBX	ColWri2.2X	Sp. 2016	54	40698
UCBX	CS169.2X	Sp. 2016	204	940
UCBX	Policy01X	Sp. 2016	129	1804

Table 2: Main Hyperparameters by Architecture Type

Architecture Type	LSTM	Transformer
Max. Seq. Length	256	256
Main Layer Width	128	128
Layer/Block Count	2	2
Attention Heads	N/A	8
Optimizer	Adam	Adam
Learning Rate	0.01	0.0005
Batch Size	128	64

- Prepend the token ID representing the course homepage to every sequence that does not already begin with this token ID, then pad or truncate of the resulting sequences to the maximum sequence length, adding 0 tokens if necessary.

2.1.2 Additional Notes on Dataset Selection

In [4], additional criteria are included for selecting courses used to demonstrate the utility of a student course trajectory model, including approximating of the entropy of each dataset as a set of discrete random processes via fitting a HMM (Hidden Markov Model) to each dataset. For the experiments in this paper, limited access to MOOC trajectory records preempts the utility of filtering out datasets with low entropy over all course sequences. Table 1 provides summary statistics for the six courses chosen for this paper, hailing from the MOOC offerings of these two universities:

- DelftX from the Delft University of Technology in Delft, Netherlands
- UCBX from the University of California, Berkeley in Berkeley, California

2.2 Hyperparameters and Training Context

All training and evaluation was completed on a remote Linux server CPU equipped with 2 GeForce Titan X GPUs (Graphics Processing Units). The script for training and evaluation is written in Python 3 using the Keras [1] deep learning API over a Tensorflow backend. Table 2 provides the full set of hyperparameters used for training and evaluating each model on each course record dataset. As the goal of this paper is to demonstrate specific differences in model architecture and training that lead to performance gains relative to the earlier results, hyperparameter tuning was not done for any of the LSTM models to facilitate comparison with results in [3]. Additionally, minimal hyperparameter tuning was done on for the Transformer models in order to minimize the risk of overfitting to the datasets for each course.

Simultaneous Fitting to Multiple Datasets. Since records from each of the 6 courses were processed as described in Section 2.1 independently, attempting to fit models on multiple courses would result in collisions between different sets of course node tokens. Nonetheless, building a predictive model that can fit to datasets from a wide range of courses is a well-defined area for future research.

2.3 Empirical Results

Table 3 presents summary statistics for each model’s final test accuracy and total training time per batch for each of the six datasets listed in Table 1. Table 4 presents additional metrics for the Transformer model pertinent to the analysis in Section 2.4.3. All statistics in both Table 3 and Table 4 are recorded using the default set of Keras command line logging tools.²

2.4 Analysis and Further Considerations

2.4.1 Baseline LSTM Comparison with Previous Results

At face value, the results for the Baseline LSTM model corroborate those presented in [4], with the caveat that average accuracy metrics reported in [4] are calculated in a different fashion that effectively gives more weight to correctly predicting tokens that occur in shorter course trajectory sequences.

2.4.2 Comparison of Final Test Accuracy Between Models

Table 3 and Table 4 show that the Transformer model achieves an average final test accuracy of around 63 percent, approximately on par with the average final test accuracy of both LSTM models without tied embeddings, in contrast with a marginally yet consistently higher 64 percent average for the LSTM models that use Tied Embeddings (as described in Section 1.5). On the other hand, including the Confidence Penalty (as described in Section 1.5) does not provide any meaningful improvement in final test accuracy for any of the six datasets. As the training scheme for all models featured in this paper invoke early stopping after 3 epochs without improving validation loss, training any of the above models for more epochs will most likely lead to overfitting on the training set.

2.4.3 Further Analysis of Final Test Accuracy for Transformer Models

At a first glance, the final test accuracy results in Table 4 seem to contradict Transformer models’ considerable performance improvements over LSTM models demonstrated in [7]. Nevertheless, the largest dataset featured in this paper only includes 40,698 course trajectory sequences, which is multiple orders of magnitude smaller than the WMT machine translation datasets used in [7] with millions of sentence pairs per language pair. This discrepancy in dataset size can cause overfitting for a particular deep learning architecture optimized to train with much larger datasets, even while controlling for model and training hyperparameters. Furthermore, the final training accuracies listed in Table 4 suggest that the Transformer model has overfit to

²BaseLogger and ProbarLogger Callback utilities. [1]

Table 3: Overall Performance Metrics by Architecture

Model	Baseline LSTM	LSTM w/ Conf. Penalty	LSTM w/ Tied Emb.	LSTM w/ Both Enh.	Transformer
Final Test Accuracy					
Average	0.6373	0.6355	0.6418	0.6388	0.6383
Std. Dev.	0.05623	0.05558	0.05728	0.05592	0.05455
Training Time per Batch					
Average	35 ms	35 ms	35 ms	35 ms	2 ms
Std Dev.	0.94 ms	0.92 ms	0.89 ms	0.86 ms	0.04 ms

Table 4: Additional Performance Metrics for the Transformer Model

	Test Acc.	Train Acc.
Average	0.6307	0.6383
Standard Deviation	0.06081	0.05455
Avg. for Large Datasets	0.6438	0.6411
Avg. for Small Datasets	0.6175	0.6355

the smaller datasets in this paper despite the use of early stopping, particularly for the following two datasets from courses with fewer than 2,000 unique users as recorded in Table 1:

- UCBX CS169.2X with 904 unique users
- UCBX Policy01X with 1,804 unique users

In conclusion, these results provide evidence that Transformer-based models do not yield benefits in accuracy over LSTM models when trained with datasets of similar size to the MOOC course trajectory datasets featured in this paper, in contrast with the much larger datasets common to certain NLP tasks such as machine translation.

2.4.4 Further Analysis of LSTM Enhancements

Given that both the Tied Embedding Layers and the Confidence Penalty are theoretically motivated by a search for new forms of model regularization, the empirical results in Table 3 indicate the Tied Embeddings are a marginally more effective form of regularization than the Confidence Penalty for this task. Furthermore, since the Tied Embedding enhancement specifically targets the input embedding and output layers of a discrete next-step predictive model for regularization in contrast to the Confidence Penalty altering the entire model’s loss function, the embedding and output layers of each of the LSTM models play a disproportionately important role in the model’s performance as a whole for this task.

2.4.5 Analysis of Training Time Results

In contrast to the Transformer model’s lack of improvement in final test accuracies for all six datasets, the results in Table 3 and Table 4 suggest that the Transformer model outperforms all types of LSTM models by more than an order of magnitude with respect to total training time per batch. Given that both the LSTM and Transformer models are built to accommodate a maximum sequence length of 256 as indicated in Table 2, the results in Table 4 are consistent with the reduced number of training passes through the

model’s computational graph per input sequence, as encapsulated in the Transformer architecture’s design goals from Section 1.4.

2.5 Directions for Future Research

2.5.1 Task-Specific Model Enhancements

As mentioned in multiple sections of this paper, certain task-specific strategies for improving performance on MOOC course trajectory prediction covered in [4] are not investigated in this paper, even if applying these strategies in conjunction impact performance in a noteworthy manner. Some of these additional strategies used to improve performance on these two tasks include:

- Calculating the entropy of each dataset’s best-fit HMM transition matrix as a criterion for selecting MOOC course trajectory datasets used to evaluate the enhanced LSTM and Transformer models.
- Incorporating auxiliary data inputs, including the time difference between course navigation actions, into evaluating the benefits of enhanced LSTM models and Transformer models over baseline results from [4].

2.5.2 Model Pre-Training and Multitask Learning

Another more ambitious goal for further research involves constructing one model that can provide meaningful predictions for multiple tasks with minimal training needs. Given the wide applicability of models with generalized predictive modeling capabilities, this model would most likely incorporate innovations originally designed to provide multitask capabilities for NLP applications. For example, [6] presents a NLP model that is first trained to perform a next-word prediction task on large text datasets before undergoing fine-tune training for more specific downstream NLP tasks, which include tasks such as text classification, sentence embedding, question answering and free-form text generation. In the context of education analytics tasks, an analogous suite of tasks for such a model could include modeling overall course performance and individualized suggestions for instructors assisting students with course material.

3. REFERENCES

- [1] F. Chollet et al. Keras. <https://keras.io>, 2015.
- [2] H. Inan, K. Khosravi, and R. Socher. Tying word vectors and word classifiers: A loss framework for language modeling. *arXiv preprint arXiv:1611.01462*, 2016.
- [3] Z. Pardos, Z. Fan, and W. Jiang. *Connectionist recommendation in the wild: On the utility and scrutability of neural networks for personalized course*

- guidance*, volume 29, pages 487–525. Springer Nature B.V., Netherlands, 2nd. edition, 2019.
- [4] Z. Pardos, S. Tang, D. Davis, and C. V. Le. Enabling real-time adaptivity in moocs with a personalized next-step recommendation framework. In *Proceedings of the Fourth ACM Conference on Learning @ Scale, L@S '17*, pages 23–32, Cambridge, Massachusetts, USA, 2017. Association for Computational Linguistics.
 - [5] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548v1*, 2017.
 - [6] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018.
 - [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkorei, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Return of the Student: Predicting Re-Engagement in Mobile Learning

Maximillian Chen
Cornell University
mlc294@cornell.edu

René F. Kizilcec
Cornell University
kizilcec@cornell.edu

ABSTRACT

Mobile learning platforms cater to intermittent microlearning by lowering the barrier for re-engaging in the learning process after a period of disengagement. We examine student re-engagement in the context of an SMS-based mobile learning platform, how to predict it and how it differs from disengagement. In a sample of 87,651 Kenyan students, we analyze data on 1,196,780 quiz attempts, finding that 36.3% of students who disengage for a week or more eventually re-engage on the platform. They spend more time on quizzes early on than students who stay disengaged. A Random Forest classifier trained on two days of student activity logs predicts disengagement and re-engagement with similar performance: F1 scores of 81.2% and 80.9%, respectively. The prevalence of re-engagement in mobile learning calls for more research into this behavioral outcome.

1. INTRODUCTION

As the world becomes increasingly connected through mobile technology, mobile devices are becoming an increasingly viable medium for education. Not only are mobile phones more affordable than traditional personal computers, but mobile devices have shallower learning curves, as they require lower levels of literacy and training [14]. The accessibility of mobile technology is especially advantageous in resource-constrained areas. It provides students access to educational resources without having to make substantial economic trade-offs associated with desktop computers. Given the rapid development of mobile computing power, many people in developing economies are predicted to skip purchasing desktop computers altogether and instead adopt mobile devices [6]. In comparison to traditional online learning platforms, mobile learning platforms remain relatively understudied despite their promise for accessibility.

A common concern with self-directed learning tools is that students do not stay engaged on the learning platform for long. The issue of disengagement, defined as a drop in student activity on the platform, has been studied extensively,

for instance in the context of Massive Open Online Courses (e.g. [16, 15, 12, 8]). However, it remains largely unstudied in the context of mobile learning environments. Student engagement patterns likely vary between desktop and mobile learning environments, considering how many different applications are available [2] and how deeply embedded mobile devices are in people's everyday lives. In fact, mobile learning platforms have been found to provide unique opportunities for microlearning sessions, where learning tasks are broken into shorter chunks that can be managed "on-the-go" [4]. Especially considering the low barriers to entry and exit in most mobile learning applications, it is unsurprising that a sizable proportion of students engage and disengage freely, which can result in longer gaps of inactivity. These intermittent usage patterns require that we consider re-engagement as a distinct behavior in mobile learning and how it compares to disengagement. Insights from this work can advance our understanding of how mobile learning works in practice and how platforms may support at-risk students through intervention.

In this research, we propose definitions of disengagement and re-engagement in mobile learning, analyze differences in behavior between disengaging and re-engaging students, and apply supervised machine learning approaches to predicting disengagement and re-engagement in mobile learning. We find that 36.36% of students who disengage for a week eventually re-engage on the platform within two weeks. A Random Forest classifier trained on two days of student log data can predict re-engagement after two weeks with an F1 score of 80.9%, showing that early platform activity is indicative of which students will return later on.

2. BACKGROUND

2.1 Beyond Student Disengagement

Before defining re-engagement, we need to formally define its prerequisite: disengagement. Defining disengagement in mobile learning platforms presents a challenge, because many such platforms are inherently less rigid and prescriptive in their learning design compared to online learning environments such as massive open online courses (MOOCs). MOOCs tend to lay out a clear path through course materials with deadlines, while many mobile learning platforms provide more room for self-directed learning and agency in choosing a learning path. This calls for an updated definition of disengagement for the context of mobile learning.

Disengagement is defined conceptually as a "lack of engage-

Maximillian Chen and Rene Kizilcec "Return of the Student: Predicting Re-Engagement in Mobile Learning" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 586 - 590

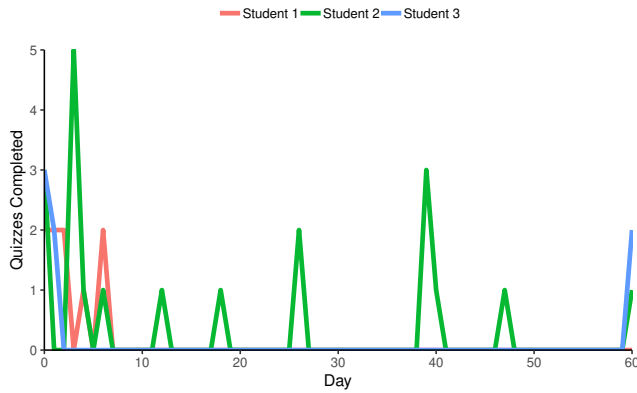


Figure 1: Three students’ daily activity for 60 days after signing up: student 1 disengages; student 2 stays engaged; student 3 disengages and re-engages.

ment,” and has been operationalized in terms of students’ interaction with or completion of learning objectives, depending on the structure of the learning platform [12]. In past studies in the context of MOOCs, disengagement has been defined as a “lack of interaction” [9, 1], the point in time where a student “fails to submit any further assignments” [15], failure to earn a course certificate [5], failure to complete a set of modules [3] or a lack of platform interaction combined with a lack of progress towards course completion [12]. Despite the many studies defining and predicting engagement in MOOCs, research on modeling engagement in mobile learning is scarce [7] and definitions of disengagement may not translate well from MOOCs to mobile learning. Definitions focusing on course completion do not apply in a context that is unlike a course, and definitions focusing on an absence of platform interaction may incorrectly label students who disengage early but return to the platform later on. Mobile learning platforms offer more opportunities for “microlearning” sessions in which students are able to learn in short bursts sporadically or “on-the-go” [4].

A recent study on engagement patterns in mobile learning on the same platform as in this research found students tend to be engaged in learning activities during the first few days after signing up but disengage shortly thereafter. In fact, 75% of all students disengaged within two days of registering, and even among the cluster of engaged students, 68% of them appeared to disengage in the first ten days [7]. However, whether a student has completely disengaged can be unclear at first sight. Consider the three actual students whose activity over time is visualized in Figure 1. All three are engaged in the first week, but student 1 disengages and never re-engages, while student 2 is inactive during the second week but occasionally returns to complete quizzes over the next two months. Student 3 was engaged on the day of registration but then disengaged for 60 days before re-engaging. We therefore define a re-engaging student as one who disengages but then returns to the mobile learning platform. This more accurately characterizes student behavior in the long run and with some additional granularity.

The ability to distinguish re-engaging students from disen-

gaging students has practical applications, such as for an automated student support system. The system could send different kinds of text messages or notifications to students who are classified as disengaging (i.e. not ever re-engaging) based on their activity in the first few days. By targeting students based on their predicted behavior, providers can tailor reminders to groups of students to highlight learning opportunities without alienating students who are already likely to re-engage in the absence of nudging.

2.2 Predicting Student Engagement

As there have not been any large-scale studies predicting student engagement in mobile learning to date, we build on a large literature on predicting student engagement and intervention systems in the context of MOOCs [16, 12, 15]. As is the case in MOOCs, a vast majority of students on mobile learning platforms eventually disengage. Any supervised learning approach in which labels correspond to engagement/disengagement would therefore suffer from class imbalance, i.e., the distribution of class labels is heavily skewed [10]. A naive classifier could simply predict the majority label for all instances and achieve a high degree of accuracy without successfully identifying actual engagement. Nagrecha and colleagues [12] addressed this issue by re-sampling their training data to balance the distribution of their labels, as was done in prior work predicting student disengagement [11]. Due to class imbalance, model accuracy can be a misleading evaluation metric, and prediction recall is frequently used as a substitute. Likewise, in this study, we face a heavy imbalance in class labels (very few students re-engage). We therefore opt to re-sample our data during training and evaluate our models using both recall and F1 score.

The user interface of mobile learning platforms tends to be simpler than ones designed for larger computer screens. For example, MOOCs tend to have more advanced platform features than mobile learning applications, such as video playback options and non-traditional assessment types. Prior studies have focused on engineering features relevant to interaction with video lectures, such as “number of straight-through video plays” or “number of video views per session” [9]. But clickstream data (interaction logs) are more informative about interactions with the structure of a platform than any specific course, which is why they were found to offer strong predictive power when analyzing data across multiple courses on a learning platform [17]. We pose two research questions in this study:

RQ1. How does the behavior of re-engaging students compare with those who stay disengaged?

RQ2. What features are predictive of student re-engagement?

3. METHODS

3.1 Platform Background & Dataset

We study re-engagement on a text message-based mobile learning platform called Shupavu 291. It has been used by over 5 million students and it offers content for over 800 distinct curricula. The platform was developed by Eneza Education¹ to provide a learning resource in regions with

¹<https://enezaeducation.com/>

Table 1: Daily student activity features for two days.

Feature Name	Definition
time.i	Time spent on day i
nlessonsfinished.i	Num. of lessons completed on day i
nask.i	Num. of questions asked on day i
nquizzes.i	Num. of quizzes completed on day i
avg_solve_time.i	Avg. time to complete quizzes on day i
n_unique_quizzes.i	Num. of unique quizzes completed on day i
nsummary.i	Num. of quiz results viewed on day i
nhw_tools.i	Num. homework tools (e.g. dictionary) used on day i

limited access to education. Shupavu 291’s user base primarily consists of Kenyan students, though its influence is growing in other African countries. The platform was designed by a group of Kenyan teachers, and the course materials align with the topics and learning outcomes of the Kenyan national curriculum for numerous subjects in primary and secondary education. Every interaction with Shupavu 291 is via text message. Students navigate through menus and quizzes by sending a text message containing a number corresponding to a menu item from the options relayed to them. Students are able to choose from a variety of grade-specific subjects such as “Fractions” and “Kiswahili.” For a given subject, students choose a specific topic and receive compact lecture notes followed by a quiz (generally five multiple-choice questions). Quiz questions follow the menu format and are sent individually; students receive instant feedback on correctness along with an explanation. Students may retake quizzes as many times as they like or use the “Ask-A-Teacher” feature to ask teachers for help.

Shupavu 291 stores a record for every quiz or platform interaction a student completes. The dataset used consists of 21,302,582 platform actions, including 1,196,780 quiz attempts, from 87,651 students in Kenya. Data beyond self-reported grade level and platform interactions for each student is completely de-identified. For the purpose of this research, we construct two sub-samples, where each one is used to solve a separate prediction problem. The first sample consists of the 87,651 students who completed at least one quiz on Shupavu 291 (an indicator of their willingness to engage with content). The second sample consists of those 63,120 students in the first sample who exhibited a seven-day period of inactivity (i.e. disengagement). The sample definitions are explained further in the next section.

3.2 Defining the Prediction Task & Features

We define two separate prediction tasks: predicting disengagement, and predicting eventual re-engagement. A disengaging student is defined as one who is inactive (here, not attempting quizzes) for at least seven consecutive days. A re-engaging student is defined as one who has disengaged and then is active (here, attempts quizzes) for at least two different days within the 14-day period following the period of inactivity. As in most disengagement prediction problems [12], we found a significant imbalance in observed labels for both disengagement and re-engagement: 72.01% of students were labeled as disengaging, and 63.68% of them were labeled as remaining disengaged (i.e. not re-engaging). We thus trained our classifiers on data that was randomly re-sampled to achieve a more balanced label distribution.

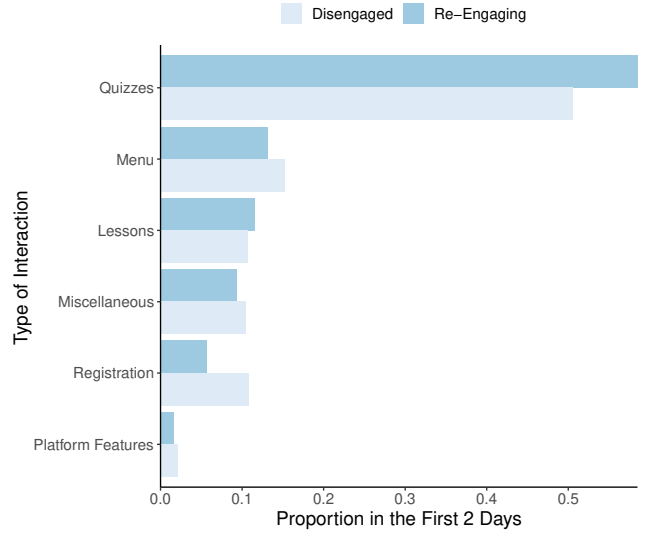


Figure 2: Distribution of the number of interactions with different parts of the platform in the first two days for students who re-engage and those who stay disengaged.

Due to the rapid decline in student engagement after registration, we devise features to capture activity on each student’s first two days on the platform. We expect early engagement to be predictive of disengagement and re-engagement. The features, defined in Table 1, capture how students interact with key components of the Shupavu 291 application and generalize across multiple subject areas, similar to the method used by Taylor and colleagues [15]. Min-max normalization is performed for each feature for each student’s first two days. We fit a Random Forest (RF) to predict whether students disengage, then another RF to predict their re-engagement. Model performance is evaluated using Recall (as suggested in [12]) and F1 scores (the harmonic mean of Precision and Recall), where a “true positive” is a student who disengages (or remains disengaged). We optimize model hyper-parameters to maximize F1 scores through exhaustive 5-fold cross-validated grid search using scikit-learn [13].

4. FINDINGS

We find that more than a third (36%) of students who disengage (seven days of inactivity) eventually re-engage on the Shupavu 291 mobile learning platform. The prevalence of re-engagement in this learning context speaks to the importance of considering this engagement pattern in mobile learning more broadly. To address the first RQ about differences between disengaging and re-engaging students, we compare student activity in the first two days after registering on the platform. Actions on Shupavu 291 are grouped into six categories:

- *Registration*: managing Shupavu 291 subscriptions.
- *Menu*: navigating the menu structure.
- *Lessons*: using course material, e.g. completing lessons.
- *Quizzes*: answering quiz questions, checking quiz grades, or starting quizzes.
- *Platform Features*: using Shupavu 291-specific resources,

e.g. the dictionary or ask-a-teacher feature.

- *Miscellaneous*: any other interaction, e.g. promotional events and features.

Overall, disengaging and re-engaging students behave similarly, spending most of their time interacting with quizzes (Figure 2). However, re-engaging students interact significantly more with quizzes (56.00% v. 47.57%, $\chi^2 = 25083, p < 0.001$) and slightly more with lessons (11.45% v. 10.34%, $\chi^2 = 1109.6, p < 0.001$, while disengaging students have more registration events (13.30% v. 7.25%, $\chi^2 = 35827, p < 0.001$). Having a greater proportion of registration events may be an indication that students who stay disengaged were already spending less time engaging with Shupavu 291 even within their first two days. A greater proportion of academic (quiz and lesson) events is likely an indication that students who eventually re-engage were more active students early on. The finding that re-engaging students engage with more academic events early on is notable, as quizzes and lessons are the core functions of Shupavu 291.

4.1 Predicting Modes of Engagement

We fit an RF² to predict disengagement and re-engagement using a set of features that capture early platform activity (Table 1). The model achieved good results for the disengagement prediction task, with a testing F1 score of 81.21% and Recall of 83.06%. Fitting the same RF to predict re-engagement received comparable performance: 80.91% F1 score and 84.19% Recall. This suggests that it is possible to train a useful classifier for both behaviors using early engagement features.

To better understand which kinds of early behaviors predict each outcome, we compare variable importance scores between the models in Figure 3. The number of quizzes completed on both day 0 and day 1, time spent on day 1, and number of platform features (questions asked, homework tools, quiz summaries) used on both day 0 and day 1 are more important for predicting re-engagement, whereas the other features are more important for predicting disengagement. This suggests that quiz engagement and diversity of platform usage is especially predictive of a student's likelihood to re-engage, though many of these characteristics are also predictive of disengagement. The importance of time spent on day 1 for predicting re-engagement is notable because it indicates that long-term behavior is related to sustained activity. Aside from the finding that diversity of platform usage is more important for predicting re-engagement, Figure 3 also suggests that specific platform feature usage (e.g. "Ask A Teacher") is not as indicative of student engagement as in prior work with MOOCs [9]. Overall, we find that early usage behavior is predictive of students' subsequent engagement pattern, which provides a basis for developing automatic interventions to better support students.

5. DISCUSSION

This study shows the prevalence of re-engagement in mobile learning. This behavioral outcome can be defined in many different ways and the optimal choice will depend on the context of the learning environment and broader goals of the

²1,000 trees, 2 samples/split min., 1 sample/leaf min., 25 tree depth limit, Gini criterion, optimized for F1 score

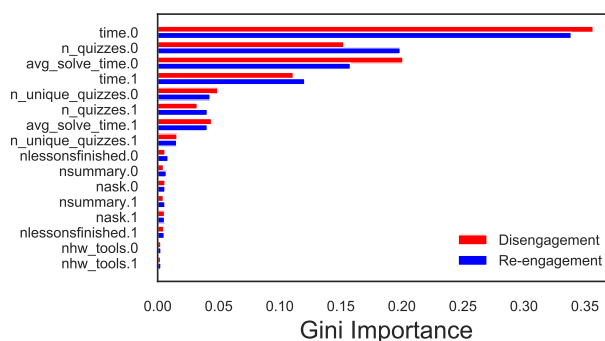


Figure 3: RF Gini Importance by prediction problem. Number of quizzes completed and time spent after the first day are more important predictors of re-engagement than disengagement.

predictive model. In particular, the periods of time and the thresholds of activity to determine dis- and re-engagement can be tweaked to fit context and goals. In the context of Shupavu 291, which has rapid disengagement, most periods of inactivity occur soon after registration. Most predictive models are not also explanatory models and this is no exception. While it is feasible to predict how a student will behave, it is unclear why they (choose to) behave in that way. A student who is active early on but disengages for a seven-day period several weeks after registering could be treated differently than one who disengages early on for the purpose of targeted intervention. Yet more work is needed to discern how to support students differentially in light of their predicted outcome.

One of the limitations of this study lies in how the second prediction problem is set up. The model is trained only on the subset of students who disengage, because by our definition, a student who does not disengage cannot re-engage. Alternatively, we could have taken the output from the disengagement prediction model and predicted the joint likelihood of disengaging and re-engaging. However, this would have introduced a great deal of uncertainty from the disengagement task into the re-engagement task. Another alternative is to set up the re-engagement prediction task as identifying students who disengage and then re-engage; however, in this case, all other students are then a mix of those who disengage completely and those who remain engaged the entire time—two groups which exhibit very different behaviors. Restricting the sample to only disengaged students gives up some information, but provides a clear basis of comparison for predicting disengaging and re-engaging students.

This research contributes an empirical treatment of student re-engagement in mobile learning and one of the first large-scale studies of student interaction with a mobile learning platform, especially in the developmental context of Sub-Saharan Africa, where mobile learning provides students with affordable access to study resources outside of formal schooling. We find it is possible to predict and distinguish between disengaging and re-engaging students using early clickstream data, providing a foundation for more research into patterns of re-engagement.

6. REFERENCES

- [1] G. Balakrishnan and D. Coetzee. *Predicting student retention in massive open online courses using hidden markov models*. Technical Report No. UCB/EECS-2013-109, 2013.
- [2] I. Blair. *Mobile App Download and Usage Statistics (2019)*, Feb 2019. <https://buildfire.com/app-statistics/>.
- [3] J. Dillon, N. Bosch, M. Chetlur, N. Wanigasekara, G. A. Ambrose, B. Sengupta, and S. K. D’Mello. Student emotion, co-occurrence, and dropout in a mooc context. In *Proceedings of the 9th International Conference on Educational Data Mining*, 2016.
- [4] T. Dingler, D. Weber, M. Pielot, J. Cooper, C.-C. Chang, and N. Henze. Language learning on-the-go: opportune moments and design of mobile microlearning sessions. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 1–12, 2017.
- [5] J. He, J. Bailey, B. I. Rubinstein, and R. Zhang. Identifying at-risk students in massive open online courses. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
- [6] G. Jones. *Clickology: What Works in Online Shopping and How Your Business Can Use Consumer Psychology to Succeed*. Hachette UK, 2013.
- [7] R. F. Kizilcec and M. Chen. (in press). Student engagement in mobile learning via text message. In *Proceedings of the Seventh (2020) ACM Conference on Learning @ Scale*, 2020.
- [8] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the third international conference on learning analytics and knowledge*, pages 170–179, 2013.
- [9] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart. Predicting mooc dropout over weeks using machine learning methods. In *Proceedings of the EMNLP 2014 workshop on analysis of large scale social interaction in MOOCs*, pages 60–65, 2014.
- [10] R. Longadge and S. Dongre. Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707*, 2013.
- [11] C. Márquez-Vera, A. Cano, C. Romero, A. Y. M. Noaman, H. Mousa Fardoun, and S. Ventura. Early dropout prediction using data mining: a case study with high school students. *Expert Systems*, 33(1):107–124, 2016.
- [12] S. Nagrecha, J. Z. Dillon, and N. V. Chawla. Mooc dropout prediction: lessons learned from making pipelines interpretable. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 351–359, 2017.
- [13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [14] I. Quadir. Form, transform, platform: How the ubiquity of mobile phones is unleashing an entrepreneurial revolution. *Innovations: Technology, Governance, Globalization*, 7(4):3–12, 2012.
- [15] C. Taylor, K. Veeramachaneni, and U.-M. O’Reilly. Likely to stop? predicting stopout in massive open online courses. *arXiv preprint arXiv:1408.3382*, 2014.
- [16] J. Whitehill, K. Mohan, D. Seaton, Y. Rosen, and D. Tingley. Delving deeper into mooc student dropout prediction. *arXiv preprint arXiv:1702.06404*, 2017.
- [17] J. Whitehill, J. Williams, G. Lopez, C. Coleman, and J. Reich. Beyond prediction: First steps toward automatic intervention in mooc student stopout. *Available at SSRN 2611750*, 2015.

Does autonomy help Help? The impact of unsolicited hints on help avoidance and performance

Christa Cody*
North Carolina State
University
cncody@ncsu.edu

Mehak Maniktala
North Carolina State
University
mmanikt@ncsu.edu

David Warren
North Carolina State
University
dswarre2@ncsu.edu

Min Chi
North Carolina State
University
mchi@ncsu.edu

Tiffany Barnes
North Carolina State
University
tmbarnes@ncsu.edu

ABSTRACT

Research has shown that autonomy can be beneficial to both learning and motivation; however, limited research has explored unsolicited hints impacts on students' autonomy. Furthermore, some research has shown that unsolicited hints can improve student learning while other research suggests that on-demand hints are more beneficial. In this study, we compare three types of student autonomy regarding hints: 1) *Control*, with on-demand hints, 2) *Choice*, with periodic popups asking whether the student would like a hint, and 3) *Assertions*, with periodic unsolicited hints. We found that the Control and Assertion groups performed similarly, and significantly better on the post-test than Choice. Further, the Assertions group had the fewest steps where help was needed but was not received, effectively solving the help avoidance problem. Overall, our results suggest that unsolicited hints can effectively ensure that more help is delivered when it is needed, reducing autonomy without reducing learning.

1 Introduction

Although research has shown that allowing students to have autonomy while learning a new domain can benefit learning [6, 18, 19, 17, 7], studies have shown that students many not have the required skills to self-regulate their learning to seek help appropriately [2, 13, 22, 12, 3, 2]. Further, research has shown that students often cannot make effective decisions regarding when they need a hint [22]. Students lacking help-seeking abilities often partake in help avoidance, where they do not use assistance available in a tutoring system [1, 15]. To address help avoidance, some ITSs employ proactive assistance [21]. While one paper found that on-demand assistance, where students have to request hints, produced

*This material is based upon work supported by the National Science Foundation under Grant No. 582690.

Christa Cody, Mehak Maniktala, David Warren, Min Chi and Tiffany Barnes "Does autonomy help Help? The impact of unsolicited hints and choice on help avoidance and learning" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 591 - 595

better learning outcomes [16], other studies have shown that providing tutor-initiated, unsolicited hints at the appropriate time, i.e. with no student autonomy about hints, can augment students' learning experience and improve performance [5, 14, 4].

The goal of this work is to investigate whether unsolicited hints can solve the help avoidance problem. We compare three groups: 1) *Control*, with on-demand hints, 2) *Choice*, where students were periodically asked if they would like a hint, and 3) *Assertions*, where unsolicited hints were periodically added to the student's workspace without any element of student choice. The Assertions group provided students with the least autonomy regarding when to receive a hint, by adding unsolicited hints to the workspace. Students may ignore these hints, but as they are the most efficient next step, students avoiding them will have less efficient solutions. The Choice group is the middle ground for hint autonomy because students can choose not to receive a hint. Due to the need to make a help-seeking decision, we consider this group to have a medium level of hint autonomy. The Control group is considered the most autonomous because they control the entire interaction surrounding hints. Overall, we hypothesize that the benefit of receiving help when it is needed outweighs the negative impact of removing student autonomy about when and whether to receive a hint.

We constructed the following hypotheses based on prior work in Deep Thought, a logic tutor, and research in students' self-regulation abilities: H_1 , Assertions will increase the chances of receiving help when it is needed, while not harming performance; H_2 , the Choice group will demonstrate more help avoidance than the Assertion group and worse performance in the posttest due to bad self-regulation choices; and H_3 , the Control group will also demonstrate more help avoidance than the Assertion group, and take longer in the training, but have similar performance in the posttest.

2 Deep Thought, our logic tutor

Our propositional logic tutor, Deep Thought, [11] presents proof problems as a set of given logic statements, shown at the top of the workspace and a conclusion to be derived at the bottom of the workspace (see Figure 1). Students solve problems by iteratively deriving new logic statement nodes until they derive and justify the conclusion. To create a

new statement node, students first ‘justify’ it by selecting 1-2 existing nodes and a rule to apply to them. The tutor is divided into an introduction, pretest, training, and posttest. The introduction includes two worked examples where students click through the derivation and justification of all the nodes, followed by one practice problem to learn the interface. Next, a student takes the pretest problem, which we use to compare the student’s incoming proficiency for stratified sampling (see Section 3). Next, the tutor guides students through the **training** section (15 problems) with varying difficulty, where students can request and receive hints. Finally, students take a more difficult non-isomorphic **posttest**, where all students must solve the same set of 4 problems without any tutor assistance. Throughout the tutor, including the pre- and post-test problems, our logic proof tutor provides immediate error feedback for rule application mistakes.

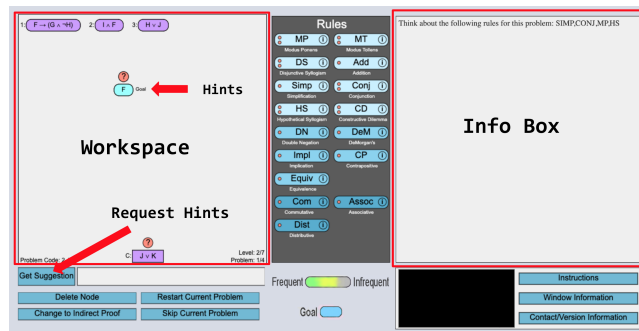


Figure 1: The Deep Thought interface.

2.0.1 Assistance

The tutor uses a data-driven approach based on a modified version of Hint Factory [20, 9] to generate hints from historical student data, resulting in hints on the most *frequent* and *efficient* paths available based on the student’s current attempt. Hints provided in the training of the tutor can either be initiated by the student, in which case they are called *on-demand* hints, or they can be initiated by the tutor, in which case they are called *unsolicited* hints. For our hints, we used our recently-designed Assertions interface [9] to place next-step hints in the workspace, which are the next, best statement that can be derived in one rule-application step from the student’s current state, as blue nodes marked with a question mark (denoting that they have not been justified) and a ‘Goal’ label. Although each group received hints through the Assertions interface for a fair comparison, later iterations of the tutor use the Assertion interface only for unsolicited hints. Hints do not tell students which rules or prior nodes can be used to justify the suggested statement and are designed to help students solve problems by suggesting a subgoal statement to help them break down multi-step problems.

3 Methods

The tutor was used as a mandatory, online homework assignment by students in an undergraduate discrete mathematics for computer scientists course (Spring 2019). For this study, we compared 94 students’ data from three conditions to investigate the impact of student-choice on performance and behavior. The three conditions were 1) **Con-**

trol, 2) **Choice**, and 3) **Assertions**. While all conditions allowed on-demand hints, they differed slightly in unsolicited help. The **Control** group represents the normal conditions in Deep Thought with no unsolicited hints. The **Choice** group was asked “Would you like a suggestion?” after completing approximately every third step to expose poor self-regulating decisions. We chose this amount to be frequent enough to be comparable to the **Assertion** group, but not distracting. The **Assertions** group received periodic unsolicited hints on approximately 40-50% of the steps to produce assistance similar to a partially worked example, or turn-taking tutor where the tutor and the student co-construct a solution to the problem.

We used stratified sampling, splitting students by pretest performance, then randomly assigning them to **Assertions** ($n = 38$), **Choice** ($n = 27$), and **Control** ($n = 29$) to ensure all conditions were balanced in incoming knowledge. The **Assertions** group was designed to have a slightly larger size to ensure sufficient data collection, and since we felt that this condition would be more beneficial to students than the **Choice** or **Control** conditions.

We used each student’s pretest **score** to measure incoming knowledge. A student’s **score** is a combination of normalized metrics for the pretest *time*, number of *steps*, and *accuracy* on a single problem, which ranks a student based on how fast, efficient, and accurate they are compared to their current peers. To investigate student’s performance, we focused on time spent solving a problem, total attempted steps, and accuracy. **Total time** is counted from the moment a problem is on the screen until it is solved by deriving and justifying the conclusion. **Total steps** in a problem include any attempt a student makes at deriving a new step, which includes both correct and incorrect steps (node derivations). **Accuracy** is the total number of correct rule applications divided by all rule application attempts. Note that the tutor is not designed or assumed to promote large improvements in accuracy, since no penalties are assigned for incorrect rule applications, even within the pre- and posttests. We focus on steps and time per problem because it is more difficult for students to learn to determine which steps to derive to achieve shorter, more efficient proofs. Whereas, learning how to apply the rules can be done by memorization and simple practice.

Data were analyzed to compare groups for the pretest, training, and posttest portions of the tutor. ANOVA with Tukey’s post hoc tests were used to examine the significance of differences in the means of the populations between pretest groups with Benjamini-Hochberg corrections. For training and posttest metrics, we applied one way ANCOVA using the pretest as a covariate. To check that the data met assumptions, we used the the Shapiro-Wilk’s W test, Levene’s test, Q-Q plots, and histograms. Data that did not meet the assumptions were transformed using log or square-root transformations, then re-inspected. Data reported in tables are before transformation for clarity. For all tables, at least marginally significant values are bolded ($p \leq 0.10$), and significant values are marked with an asterick (* for $p \leq 0.05$).

4 Results & Discussion

This section discusses the comparison between the **Assertion**, **Choice**, and **Control** groups, and the differences in per-

formance between students.

4.1 Hint Usage and Help Need

To understand each group’s utilization of hints, we examined hint-related metrics. **# Hint Requests** is the total number of hints requested in training. **Hints Received** is the total hints a student received during the tutor, unsolicited and requested. For the Choice group, **# Asked** represents how often they were asked if they would want a hint. **Hint Justification** rate is the percentage of hints received that students connected to their current solution through justification. Table 1 shows the mean, standard deviation and Tukey HSD’s results for the hint metrics. ANOVA showed a significant difference in the mean # Hints Received ($F(2, 91) = 25.576, p < 0.01$) between the groups. Tukey Contrasts analysis showed significant differences among each comparison (Control-Choice ($p < 0.01$); Choice-Assertions ($p < 0.01$); and Control-Assertions ($p < 0.01$)). We expected these differences because the Assertions group was given frequent, unsolicited hints, the Choice group was asked if they wanted a hint at a slightly lower frequency and was only given a hint if they selected ‘Yes’, and the Control group received hints only upon request. Since all three groups could request on-demand hints in addition to any the tutor might provide or offer, we compared # Hints Requests, but there were no significant differences between the 3 groups on this metric ($F(2, 91) = 0.1816, p = 0.83$)).

Table 1: Mean and Standard Deviation(SD) of the Hint Usage Metrics in the Training.

	Control <i>n</i> = 29	Choice <i>n</i> = 27	Assertions <i>n</i> = 38
<i>Metric</i>	<i>Mean(SD)</i>	<i>Mean(SD)</i>	<i>Mean(SD)</i>
# Asked	34(10)	-	-
# Hints Received	19(16)*	35(25)*	51(12)*
# Hint Requests	21(21)	26(27)	25(23)
Hint Justification Rate	85%(25)	80%(20)*	84%(6.5)*

The Control group justified 85% of the requested hints, on average, which makes sense as students are more likely to use the hints they request [16]. The mean Hint Justification rates were 84% for Assertions and 80% for the Choice group. ANOVA results revealed a significant difference between groups for the Hint Justification Rate ($F(2, 91) = 6.0633, p < 0.01$)). Tukey Contrasts analysis showed significant differences among Control-Choice ($p = 0.03$), and Control-Assertions ($p < 0.01$)), but no significant difference between Choice and Assertions group ($p = 0.79$). This is surprising because we expected the Choice group to have a higher Justification rate than the Assertions group, since, similar to the Control, they chose to get a hint. These results suggest that unsolicited Assertions were just as well received as hints offered as a choice.

Further, we defined measures to address all three hypotheses concerning hint usage: help need, hint abuse, unnecessary hints, and steps in which they received an appropriate level of help (i.e. received a hint when needed and did not receive a hint when not needed). An important goal of this study was to investigate whether periodic unsolicited hints could address help avoidance by increasing the number of times students who needed help received it. Since our hints are partially-worked steps and students could easily ignore them, unsolicited hints should not harm students who do

not need them. We determined when a hint was needed vs. not needed via our new Help-Need model described in [8, 10]. The model uses (1) the quality of the current step based on a combined productivity measure of the optimality of their current state (how close it is to the solution based on the Hint Factory [20]), and the time taken to derive it, and (2) a prediction of whether help is needed in the next step (e.g. if the next step is not predicted to be productive, then help is needed). We note that our help-need predictor is not ground truth, but our cited work shows that the Help-Need predictor is correlated with post-test performance. **% Help Needed** is the percentage of total steps our Help-Need model identified as unproductive, where a student could have benefited from a hint, and a hint was not received **% Hint Abuse** is the percent of total steps where our model predicted no Help-Need but a student *requested* a hint, representing a bad help-seeking decision. **% Unnecessary Hint** is the percent of total steps where students *received* a hint on a step where we predicted no Help-Need, including both help abuse requests and the number of times hints were given but not needed. We also included Help Abuse because we wanted to ensure none of the conditions were promoting gaming the system. **% Appropriate Hint** is the percent of steps where Help-Need model aligned with the student need (e.g. a student received a hint when they were predicted to need one or a student did not receive a hint and the model labelled the step as no help-needed).

Table 2 shows the differences in these metrics between the groups. With ANCOVA, controlling for the pretest, we found a significant difference between the groups for % Unnecessary Hints ($F(2, 91) = 38.35, p < 0.01$) and % Help Needed ($F(2, 91) = 10.11, p < 0.01$). For % Unnecessary Hints, Tukey Contrasts analysis revealed significant differences between all 3 groups: Choice-Control ($p = 0.01$), Choice-Assertions ($p < 0.01$), and Control-Assertions ($p < 0.01$). For % Help Needed with the same procedure, we found significant differences between Choice-Assertions ($p = 0.01$) and Control-Assertions ($p < 0.01$); however, there was no significant difference in Control-Choice ($p = 0.45$). There were no significant differences for Hint Abuse ($F(2, 91) = 0.04, p < 0.96$) or the Appropriate Hint metrics ($F(2, 91) = 0.57, p < 0.56$).

Table 2: Mean and Standard Deviation(SD) of the Help Need Metrics in the Training.

	Control <i>n</i> = 29	Choice <i>n</i> = 27	Assertions <i>n</i> = 38
<i>Metric</i>	<i>Mean(SD)</i>	<i>Mean(SD)</i>	<i>Mean(SD)</i>
% Help needed	20(12)	16(11)	10(8)*
% Hint Unnecessary	4(5)*	7(5)*	15(4)*
% Help abuse	7(6)	9(9)	7(7)
% Appropriate Hint	72(11)	71(12)	73(7)

The Control group had the lowest percentage of steps with Unnecessary hints, which was expected since they had full autonomy and requested fewer hints than the other groups. The Control group also had the highest percentage of steps where Help-Need was detected, meaning that these students spent more time in steps being unproductive. The Choice group fell in the middle for both % Help Needed and % Unnecessary Hints. H_2 , stated that the Choice group would have more help avoidance than the other two groups. The Control group showed similar help avoidance to the Choice

group by not requesting hints when needed. However, the Choice group had a significantly higher Help Avoidance than the Assertion group, which provides partial evidence in support of H_2 . Additionally, the Control group having a significantly higher % Help Needed partially supports H_3 , in which we hypothesized that the Control group would not request hints often enough. The Assertions group decreased steps where students needed help but were not receiving it, confirming H_1 . Although more unnecessary hints were provided, our goal was to reduce students being stuck in steps without receiving help, which was achieved even though the frequency of unsolicited hints was not based on an intelligent policy. Incorporating an intelligent policy to determine when to give a hint should result in an even smaller percentage of help need and reduce instances of unnecessary hints. To test whether the larger percentages of Unnecessary Hints would be worse for posttest performance, a simple linear regression was calculated to predict the posttest score based on the % Unnecessary Hints and was not significant ($F(1, 91) = 0.33, p = 0.57$). Therefore, we do not believe these Unnecessary Hints had a significant impact on performance. Another simple linear regression was calculated to predict the posttest score based on the % Help Needed, and a significant regression was found ($F(1, 91) = 8.49, p < 0.01$) providing support that addressing help need is important.

4.2 Evaluating Students' Performance Across the Tutor

To examine the effects on performance each group had, the pretest and posttest performance metrics for the 3 groups were analyzed (see Table 3). ANOVA was performed on pretest metrics to determine if there was a similar distribution of proficiency between the groups. There were no significant differences between the groups on Total Time ($F(2, 91) = 0.28, p = 0.76$) or Total Steps ($F(2, 91) = 1.01, p = 0.37$) in the pretest metrics. There was a marginally significant difference between the groups for accuracy ($F(2, 91) = 2.38, p = 0.09$), but this is not a meaningful difference due to the few number of steps in the pretest and the Choice's group lower average number of steps. Therefore, we concluded that each group had a distribution of students' with similar incoming proficiency.

For the training and posttest performance metrics, ANCOVA was used controlling for pretest metrics. There were no significant differences between any performance metric in the training portion of the tutor (Total Time ($F(2, 90) = 2.07, p = 0.13$); Total Steps ($F(2, 90) = 1.84, p = 0.16$); Accuracy ($F(2, 90) = 1.34, p = 0.27$)). The posttest metrics show a significant difference in the Total Time ($F(2, 90) = 5.24, p < 0.01$) between the groups. Tukey Contrast analysis revealed that there was a significant difference between the Assertion and Choice group ($p < 0.01$); however, there was not a significant difference between the Choice and Control ($p = 0.29$) or the Assertion and Control ($p = 0.19$). There was no significant difference between the Total Steps ($F(2, 90) = 2.09, p = 0.13$) or the Accuracy ($F(2, 90) = 0.05, p = 0.95$) between the groups.

These results provide support for H_1 that the students in the Assertions group would perform similarly to the Control group; however, the Control group did not perform worse in the training as expected in H_3 . These results along with the results in 2 confirm H_1 . Assertions reduced help need without harming performance. These results provide evidence

Table 3: Pretest, Training and Posttest performance metrics for the Assertion, Choice, and Control groups.

Test	Metric	Control <i>n</i> = 29	Choice <i>n</i> = 27	Assertion <i>n</i> = 38
		Mean(SD)	Mean(SD)	Mean(SD)
Pretest	Total Time (min)	5.8(7)	4.0(2)	6.5(6)
	Total Steps	15(30)	9(7)	11(13)
	Accuracy	40(14)	35(14)	43(17)
Training	Total Time (min)	137(50)	114(49)	122(62)
	Total Steps	374(126)	348(124)	323(118)
	Accuracy	63%(12)	66%(11)	66%(10)
Posttest	Total Time (min)	37(29)	43(34)*	34(20)*
	Total Steps	104(56)	129(75)	102(47)
	Accuracy	69%(12)	69%(11)	69%(11)

in support of H_2 ; however, these results do not address why the Choice group performed worse. One theory is that the students could have been making poor self-regulated decisions, supported by Table 2, which may have made them perform worse than the Control even though they both had a choice. The prompts may have lead to the Choice group to make more help-seeking decisions than the Control, where students would have thought about hints less. However, the questions asking whether or not they would like a hint could have also been frustrating or distracting. This distraction could have caused them to lose focus; however, we would have expected the total time in the training to be significantly different in that case.

Lastly, one of our concerns was whether students were better at self-regulating than a random proactive policy. The Assertions group was the slowest in the pretest, but they were the fastest in the posttest, shown in Table 3. Their overall hint Justification rate was also high, shown in Table 1. Along with the results confirming H_1 in the Table 2 and Table 3, these results suggest that the Assertions group with unsolicited, tutor-initiated hints did no harm to students in terms of learning outcomes compared to the Control group and produced better learning outcomes than the Choice group. Therefore, these results suggest that proactively adding hints at the very least did no harm.

5 Conclusion

This work contributes an investigation of the effects of three groups with varying levels of autonomy of assistance on learning outcomes and metrics to evaluate hint usage and hint avoidance. The three groups from most autonomous to least: 1) Control, where students could request on-demand hints, 2) Choice, where students were periodically asked if they would like a hint, and 3) Assertions, where hints were periodically added to the student's workspace without any element of student choice. This study sought to determine whether students' autonomy over when and how the interface provides hints affects hint utilization and, in turn, overall success. Our results show that the Assertion and Control group produce similar learning outcomes; however, the Choice group performed worse on the posttest. Overall, our results suggest that unsolicited hints can effectively ensure that more help is delivered when it is needed, reducing autonomy without reducing learning. These results demonstrate that with an effective, machine-learned proactive hint policy, better learning outcomes are possible.

6 References

- [1] Aleven, V., McLaren, B., Roll, I., Koedinger, K.: Toward tutoring help seeking. In: International Conference on Intelligent Tutoring Systems, pp. 227–239. Springer (2004)
- [2] Aleven, V., McLaren, B., Roll, I., Koedinger, K.: Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education* **16**(2), 101–128 (2006)
- [3] Azevedo, R., Cromley, J.G.: Does training on self-regulated learning facilitate students’ learning with hypermedia? *Journal of educational psychology* **96**(3), 523 (2004)
- [4] Bartholomé, T., Stahl, E., Pieschl, S., Bromme, R.: What matters in help-seeking? a study of help effectiveness and learner-related factors. *Computers in Human Behavior* **22**(1), 113–129 (2006)
- [5] Bunt, A., Conati, C., Muldner, K.: Scaffolding self-explanation to improve learning in exploratory learning environments. In: International Conference on Intelligent Tutoring Systems, pp. 656–667. Springer (2004)
- [6] Burton, R.R., Brown, J.S.: An investigation of computer coaching for informal learning activities. *International journal of man-machine studies* **11**(1), 5–24 (1979)
- [7] Katz, I., Assor, A.: When choice motivates and when it does not. *Educational Psychology Review* **19**(4), 429 (2007)
- [8] Maniktala, M., Barnes, T., Chi, M.: Extending the hint factory: Towards modelling productivity for open-ended problem-solving. In: Proceedings of the 13th International Conference on Educational Data Mining (2020)
- [9] Maniktala, M., Cody, C., Barnes, T., Chi, M.: Avoiding help avoidance: Using interface design changes to promote unsolicited hint usage in an intelligent tutor. *International Journal of Artificial Intelligence in Education* (2020 (under review))
- [10] Maniktala, M., Cody, C., Isvik, A., Lytle, N., Barnes, T., Chi, M.: Extending the hint factory for the assistance dilemma: A data-driven proactive helpneed model and intervention to improve problem solving. *Journal of Educational Data Mining* (2020 (under review))
- [11] Mostafavi, B., Barnes, T.: Evolution of an intelligent deductive logic tutor using data-driven elements. *International Journal of Artificial Intelligence in Education* **27**(1), 5–36 (2017)
- [12] Peña, A., Kayashima, M., Mizoguchi, R., Dominguez, R.: Improving students’ meta-cognitive skills within intelligent educational systems: A review. In: International Conference on Foundations of Augmented Cognition, pp. 442–451. Springer (2011)
- [13] Price, T.W., Liu, Z., Cateté, V., Barnes, T.: Factors influencing students’ help-seeking behavior while programming with human and computer tutors. In: Proceedings of the 2017 ACM Conference on International Computing Education Research, pp. 127–135. ACM (2017)
- [14] Puustinen, M.: Help-seeking behavior in a problem-solving situation: Development of self-regulation. *European Journal of Psychology of education* **13**(2), 271 (1998)
- [15] RANGANATHAN, R., VANLEHN, K., VAN DE SANDE, B.: What do students do when using a step-based tutoring system? *Research & Practice in Technology Enhanced Learning* **9**(2) (2014)
- [16] Razzaq, L., Heffernan, N.T.: Hints: is it better to give or wait to be asked? In: International Conference on Intelligent Tutoring Systems, pp. 349–358. Springer (2010)
- [17] Schank, R.C., Farrell, R.: Creativity in education: A standard for computer-based teaching. Tech. rep., YALE UNIV NEW HAVEN CT DEPT OF COMPUTER SCIENCE (1987)
- [18] Schwartz, J.: Intellectual mirrors: A step in the direction of making schools knowledge-making places. *Harvard Educational Review* **59**(1), 51–62 (1989)
- [19] Shute, V., Glaser, R., Raghavan, K.: Inference and discovery in an exploratory laboratory. Tech. rep., PITTSBURGH UNIV PA LEARNING RESEARCH AND DEVELOPMENT CENTER (1988)
- [20] Stamper, J., Barnes, T., Lehmann, L., Croy, M.: The hint factory: Automatic generation of contextualized help for existing computer aided instruction. In: Proceedings of the 9th International Conference on Intelligent Tutoring Systems Young Researchers Track, pp. 71–78 (2008)
- [21] Vanlehn, K.: The behavior of tutoring systems. *International journal of artificial intelligence in education* **16**(3), 227–265 (2006)
- [22] Zhou, G., Lynch, C., Price, T.W., Barnes, T., Chi, M.: The impact of granularity on the effectiveness of students’ pedagogical decisions. In: CogSci (2016)

An EDM-based Multimodal Method for Assessing Learners' Affective States in Collaborative Crisis Management Serious Games

Ibtissem Daoudi
ENSI, COSMOS Laboratory
University of Manouba

ibtissem.daoudi@ensi-uma.tn

Erwan Tranvouez
LIS Laboratory
Aix-Marseille Université

erwan.tranvouez@lis-lab.fr

Raoudha Chebil
ENSI, COSMOS Laboratory
University of Manouba

raoudha.chebil@ensi-uma.tn

Bernard Espinasse
LIS Laboratory
Aix-Marseille Université

bernard.espinasse@lis-lab.fr

Wided Lejouad Chaari
ENSI, COSMOS Laboratory
University of Manouba

wided.chaari@ensi-uma.tn

ABSTRACT

Recently, Crisis Management Serious Games (CMSG) have proved their potential for teaching both technical and soft skills related to managing crisis in a safe environment while reducing training costs. In order to improve learning outcomes insured by CMSGs, many works focus on their evaluation. Despite its great interest, the learner emotional state is often neglected in the evaluation process. Indeed, negative emotions such as boredom or frustration degrade the learning quality since they frequently conduct to giving up the game. This research addresses this gap by combining gaming and affect aspects under an Educational Data Mining (EDM) approach to improve learning outcomes. Therefore, we propose an EDM-based multimodal method for assessing learners' affective states by classifying data communicated in text messaging and facial expressions. This method is applied to assess learners' engagement during a game-based collaborative evacuation scenario. The obtained assessment results will be useful for adapting the game to the different players' emotions.

Keywords

Serious game, crisis management, assessment, educational data mining, affective states, multimodal emotion detection.

1. INTRODUCTION

Recently, Serious Game (SG) development and usage have increased to improve learning benefits and to increase learners' motivation [1]. SGs applications reach out several domains such as crisis management, education, ecology, and health-care [2]. Indeed, collaborative Crisis Management Serious Games (CMSG) have proved their potential for teaching concepts related to managing different types of crisis situations such as natural disasters (earthquakes, floods), man-made disasters (terrorist

attacks, pollution), and technological crises (industrial accidents, cyber attacks) in a fun way while reducing training cost and saving time [3].

Despite its obvious interest, the exploitation of the SG concept in learning processes is not always a guarantee of its effectiveness [4]. As any learning systems, SGs rely on the implicit alignment of the learning outcomes (knowledge or skills) and the game experience (engagement, motivation). In particular, the effectiveness of a collaborative CMSG depends on different learners' characteristics including cognitive, emotional and social aspects [4]. Consequently, there has been a lot of research focused on the evaluation of SGs and their effectiveness for Crisis Management (CM) training varying in terms of crisis situation, number of players, key indicators or characterization of learners [5,6,7,8,9,10]. However by studying the state of the art, we have noticed that there is a considerable lack of studies integrating the concept of affective computing, especially learners' affective states, in the evaluation process within collaborative CMSGs [4]. Besides, most of existing works use explicit techniques for analyzing learners' behaviors during playing like pre/post questionnaires, interviews and debriefing sessions. These techniques represent a subjective evaluation that relies on non-exhaustive players' opinions and disrupts the high level of engagement provided by the game; impacting thus negatively the accuracy of evaluation results [30]. So, improving players' engagement (and thus learning outcomes) requires detecting and assessing such emotional states in a non-intruding way [11].

In this paper, we focus on addressing this gap. In doing so, we focus on the detection and analysis of learners' emotions expressed in textual and visual data to infer Flow game-play experience indicator (also called engagement) in collaborative CMSGs. To the best of our knowledge, players' engagement measure and impact on learning outcomes have not been investigated in such context. Hence, our contribution is to propose an emotion-based EDM method able to:

- 1) Assess the temporal dynamics of learners' affective states during a game-based session for CM training.
- 2) Evaluate their final states at the end of training process by classifying data communicated in text messaging and facial expressions.

Ibtissem Daoudi, Erwan Tranvouez, Raoudha Chebil, Bernard Espinasse and Wided Lejouad Chaari "An EDM-based Multimodal Method for Assessing Learners' Affective States in Collaborative Crisis Management Serious Games" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 596 - 600

- 3) Explore the final individuals' affective profiles to generate the group emotion at a global level.

The rest of this paper is organized as follows. Section 2 presents the proposed EDM-based multimodal method for learners' affective states assessment. Section 3 reports the application of our method on a collaborative CMSG used as a case study. Section 4 discusses our major findings. Section 5 summarizes the paper and presents our plans for future work.

2. AN EDM-BASED MULTIMODAL METHOD FOR ASSESSING LEARNERS' AFFECTIVE STATES

Our aim is to develop an automatic method for assessing learners' affective states (*engagement, frustration, confusion, and boredom*) using facial expressions and text analysis in collaborative CMSGs. To reach this objective, we need to perform five main steps corresponding to specific tasks namely *data collection, data annotation, data fusion, data analysis, and data visualization* as illustrated in Figure 1.

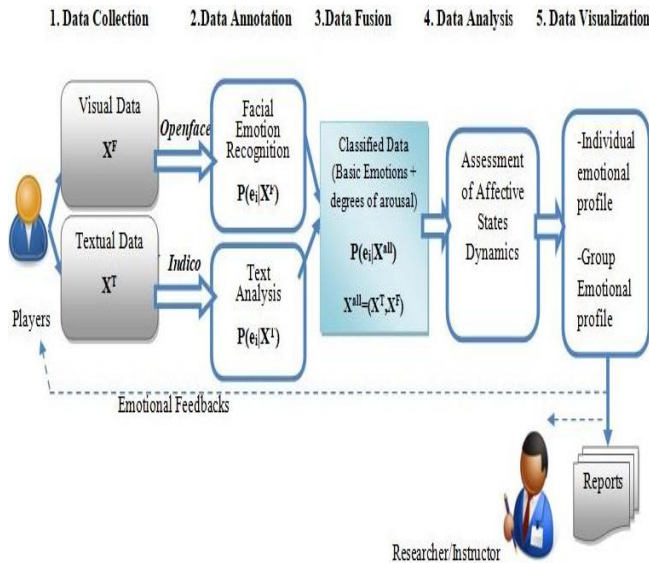


Figure 1 General Overview of the Proposed Method

2.1 Data Collection and Annotation Task

In order to collect data in a way that is more efficient and less intrusive compared to physiological measurements [12], we plan to extract the *messages* exchanged between players as well as the *video records* of learners' faces produced in real-time using a Webcam. These two kinds of data are annotated as follows:

- **Text annotation.** The textual content of these messages represents a rich source to detect their emotions that are revealed by the annotator tool *indico.io*. It is a predictive analytics tool classified as one of the top AI APIs for emotion detection from raw text strings (shorter instances of text like conversations) using deep learning algorithms with 93.5% of accuracy [13]. The API gives as an output the probability that the text reflects the basic emotions as well as their intensities.
- **Video annotation.** We have adopted *Openface 2.0*[14]: an automatic facial behavior analysis and understanding toolkit. Openface returns intensity and presence for each Facial Action Unit (FAU) estimated with several computer vision and machine learning algorithms. We

exploit the output of FAU recognition system since it displays emotions according to [15]. Based on the EMotional Facial Action Coding System [16], mapping rules associate couples of FAU with basic emotions. For example, *joy* is associated with detection of *Cheek raiser* FAU and *Lip corner puller* FAU.

2.2 Data Fusion Task

In our study, we perform a multimodal fusion at the decision-level which refers to the process of combining data collected from many modalities after being pre-classified independently to obtain the final classification. In fact, each classified modality, using the previous annotators, provides one hypothesis on labeled emotion categories; and this integration method gives a global estimate based on partial results [17].

$X^{all} = (X^T, X^F)$ represents the global feature vector consisting of the text feature vector, X^T , and the face feature vector, X^F .

In decision-level fusion, two separate classifiers provide the posterior probabilities $P(e_i|X^T)$ and $P(e_i|X^F)$ for text and face, respectively, having to be combined into a single posterior probability $P(e_i|X^{all})$; where e_i represents one of six possible classes of basic emotions ($e_1=joy$, $e_2=sadness$, $e_3=surprise$, $e_4=anger$, $e_5=fear$ and $e_6=disgust$).

The face modality is assumed to be the main modality in our multimodal approach (but the text modality is not neglected). Hence, we assign weights as follows: $\mu_T=0.3$ for the text modality and $\mu_F=0.7$ for the face modality. We adopt this weighting proposed and validated by works referenced by [18] and [19]. Then, we apply the averaging formula using these weights in order to compute the average probability of the two modalities defined as follows [20]:

$$P(e_i|X^{all} = X^T \text{ and } X^F) = (\mu_T * P(e_i|X^T) + \mu_F * P(e_i|X^F)) / 2$$

2.3 Data Analysis Task

In this task, we perform a fine-grained analysis of the dynamics of learners' affective states based on facial features during playing by studying the impact of stress on affective transitions, and we produce a summative evaluation of their emotional states at the end of training process:

- **Stress detection.** The stress is one of the most frequently occurring emotions inherent to CM since it affects the actors' way to manage crisis situations [4]. Given stress is related to emotions; also facial expressions have been used to detect stress by linking some of basic emotions as features [21]. In fact, many works have proved that, in different contexts like driving and working environments, stress is detected if either anger, fear, or a combination of these two negative emotions is detected constantly within a fixed time interval [22,23]. In particular, they focus on some specific FAU and their activation level extracted in each video frame, described as an indicator for fear and/or anger.
- **Mapping between affective states and basic emotions.** Affective states are particular combinations of basic emotions as demonstrated by [24] using association rules mining. In our study, we adopt the existing mapping as described in [24, 19]; and we propose some novel interpretations of basic emotions combinations

allowing us to deduce affective states based on existing theories of emotions [15,26].

Flow/engagement is defined by a high level of *surprise* and a low *sadness* level [19]. Since joy and sadness are opposite emotions as validated by [15] and flow is characterized by a full involvement and enjoyment in the activity [26], we can affirm that *flow* can be defined also by a high level of *surprise* and a high *joy* level.

Frustration is detected at the presence of a high degree of *anger* and a low degree of *joy* [19]. Likewise, frustration can be defined by a high level of *anger* and a high *sadness* level. Moreover, basing on the definition of frustration state [26], it can be mapped to a high level of fear as well as a high level of sadness. In the same way, frustration can be defined by a high level of *fear* and a low level of *joy*.

Boredom can be mapped to a high level of *disgust* as well as a low level of *joy*. In the same manner, *boredom* can be defined by a high level of *disgust* and a high *sadness* level [19].

The state in which *all the levels of six basic emotions* are low will represent the *confusion* affective state [19].

2.4 Data Visualization Task

This final task concerns visualization of our analysis results at two levels: individual and global. On the one hand, we visualize the summative individual emotional profiles which contain relevant information about affective states expressed by each player at the end of training process by selecting the dominant and the most pronounced emotion. On the other hand, we visualize the aggregation of all individual emotional profiles based on a decision tree algorithm to decide on the *polarity* of global emotion (*positive* or *negative*) and then to constitute the group emotion [27]. So, we apply *J48 decision tree classifier*, an implementation of *C4.5 algorithm* in *Weka*, to generate a decision tree on the group emotion based on individual affective states with a default confidence value=0.25. The principle is to decide the class label of group emotion (*positive* or *negative*) by learning decision rules inferred from training data (rates of individuals affective states). According to our experimental results, this tree-based method reaches an accuracy of 81% using *5-fold cross-validation*. Figure 2 shows the decision tree model of group emotion.

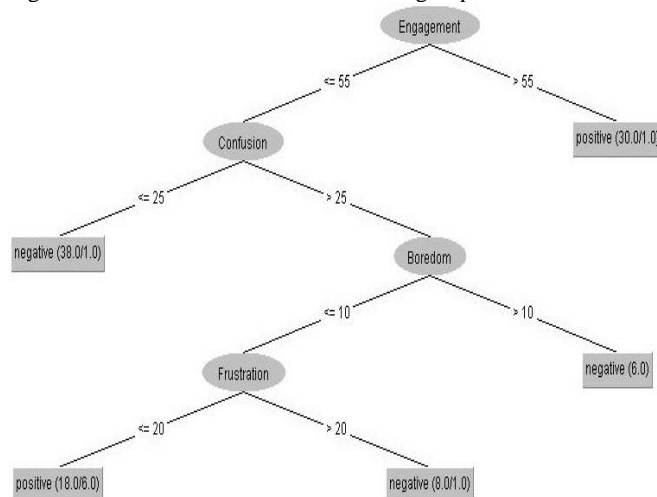


Figure 2 Decision Tree Model of Group Emotion

3. EXPERIMENT AND RESULTS

3.1 Game Description

We have developed a collaborative scenario for building evacuation training in case of a fire emergency situation. This scenario is implemented on the iScen software platform [29], specifically intended for crisis simulation, management and training. The scenario aims to train people (staff or students) of a Tunisian university building on evacuating all the present persons during a fire emergency triggered in the coffee shop as shown in Figure 3. The evacuation exercise involves a group of 30 participants (including player and virtual characters) having different roles namely *coordinator*, *security responsible*, *firefighter*, *warden* and *deputy* who must collaborate and coordinate their actions in order to manage an emergency evacuation procedure. This scenario allows learners to reach two main pedagogical objectives consisting of: (1) acquiring personal fire safety skills both in general and specifically in a university context, and (2) teaching best evacuation practices required to manage any fire emergency in an efficient and rapid manner.

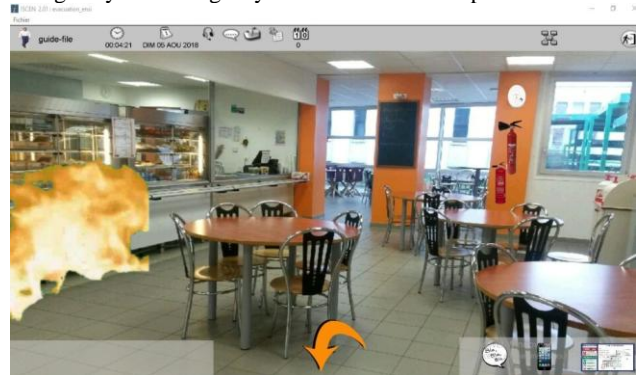


Figure 3 Screen Capture of Crisis Situation

3.2 Dataset

We analyze (n=30) students' behavior trace interaction data obtained after the game session that lasts approximately 25 minutes. These data are video recordings of the participants' faces while playing using webcams as well as exchanged messages using the text chatting system. We analyze affective dynamics experienced by participants by tracking emotions at a fine-grained level using facial features. We make judgments on what affective states were present in each 20-second interval basing on the mapping described above. In addition, pre- and post-test questionnaires were completed individually by all students before starting the game session (pre-test) and immediately after finishing it (post-test). Pre-test questions address personal information concerning prior game experience and CM knowledge. Learners are then categorized as *novice*, *intermediate* and *expert* to be confronted afterward to the experimental results. Post-test questions aim to measure the level of engagement based on the Game Engagement Questionnaire. Both pre-test and post-test are on a 5-point Likert scale ranging from 1 (not at all) to 5 (extremely).

3.3 Obtained Results

Comparing to several predictions proposed by the Cognitive Disequilibrium Model [24], it appeared that some of these predictions have been validated while others not addressed by the model are identified by our method. This model addresses transitions between affective states of learners while solving complex activities in relatively short learning sessions [24].

The supported predictions include the transitions from the state of engagement into confusion, confusion into frustration, and frustration into boredom which naturally occurred. In fact, analyzing transitions between affective states are so important because they provide insight into how learners enter into an affective state since engagement and confusion is correlated with higher performance, while frustration and boredom are correlated with poorer performance.

The two predictions that have been identified, but were unexpected in the model, include the transitions from frustration to confusion and boredom to frustration. First, even though the transition from frustration into confusion occurred rarely, we believe that some frustrated participants, could view the situation as a challenge and become more energized; and ultimately enter the confusion state while trying to resolve the current misunderstanding. Second, the transition from boredom into frustration occurred significantly when we detect a high activation level of some FAU characterizing the stress emotion. To resume, our findings suggest that some aspects of the cognitive disequilibrium model might need refinement and some transitions can occur due to a specific characteristic of the context of CM training namely the stress.

When aggregated across the all participants at the end of training process, our results indicated that 25% of learners felt engagement, 50% expressed boredom, 25% felt frustration, and 0% experienced confusion. Figure 4 displays a global view of the all individual affective states. This global view allowed us to decide the polarity of group emotion by applying our decision tree model. Hence, we can deduce that the global emotion is negative ($25\% \text{ engagement} + 0\% \text{ confusion} + 50\% \text{ boredom} + 25\% \text{ frustration} \Rightarrow \text{negative class}$).

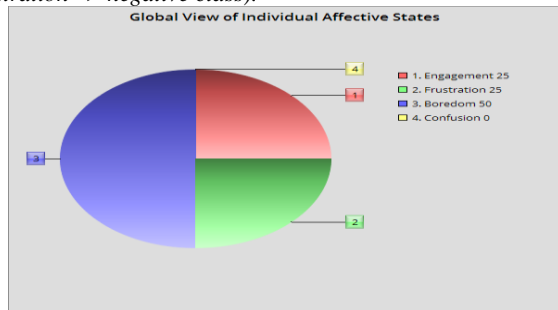


Figure 4 Global View of Individual Affective States of Players

3.4 Results Validation

For validation purpose, summative evaluation results are compared to the answers to the Game Engagement Questionnaire (GEQ) analyzing self-reported subjective descriptions and collected after the game session. This questionnaire is one of the most commonly used self-report questionnaires in the player experience field for measuring engagement specifically elicited while playing games [28]. The core module of GEQ is a 33-item scale which is designed to measure game players' experience across seven dimensions namely *Immersion*, *Flow*, *Competence*, *Positive Affect*, *Negative Affect*, *Tension*, and *Challenge*. Dimension scores are computed as the average value of their items. The descriptive statistics obtained from learners responses are reported in Table 1.

Table 1 Descriptive statistics for dimensions of the GEQ

Dimension	Mean	Standard deviation	Max	Min
-----------	------	--------------------	-----	-----

Immersion	2.13	0.60	4.00	1.00
Flow	2.34	0.62	3.55	1.13
Positive affect	2.00	0.50	3.00	1.00
Negative affect	4.28	0.85	5.00	2.00
Tension	3.96	0.77	5.00	1.65
Challenge	3.43	0.68	5.00	2.00
Competence	2.43	0.56	3.00	1.20

4. DISCUSSION

As shown in Table 1, positive feelings are much less severe and less frequently experienced compared to negative feelings (lower than the mid-value of the scale). In fact, participants reported the level of *positive affect* to be low (2.00). More specifically, results analysis shows that *immersion* (reflecting how players felt strongly connected with the game) and *flow* (indicating whether players lost track of their own effort and/or the passage of time during the game) receive respectively average degrees (2.13 and 2.34). The dimension *negative affect* receives the highest value of all (4.28). This result indicates that playing the game engendered some negative emotional experiences in particular boredom. In addition, participants experience a certain high degree of *tension* (3.96) in the form of specific negative emotions like frustration. Moreover, in terms of *challenge*, participants report that the game environment is difficult and challenging (3.43) according to their level of *competence* (2.43). All these results confirm the negative group emotion detected after the application of our method on the same CM scenario. Basing on this result, we can conclude that the team performance is also negative. In fact, this interpretation can be explained by the fact that all participants are situated, for the first time, in an emergency evacuation procedure based on a virtual training environment. It can also be a consequence of limited learners' guidance and assistance carried out by the instructor during the training process in order to better achieve the game objectives.

To summarize, the final affect annotations obtained via our method correlate well with subjective responses to the GEQ. In comparison to the GEQ, our method represents an objective and rapid manner to analyze learners' emotions and to infer their affective states without distracting them from game-play using EDM techniques. Hence, our contribution is intended to support learning, maintain motivation, and increase learners' engagement in the virtual world of the game.

5. CONCLUSION AND FUTURE WORK

This paper investigates a multimodal learner analytics approach to assess emotional states in collaborative CMSGs. Specifically, decision tree models were trained to predict learners' affective states utilizing bimodal data including textual messages and facial expressions. Affective states predicted by the model are evaluated with learners' self-reported engagement scores reported after the game session. In future work, we want to extend this study to a larger sample of participants within another multi-players CMSG which is currently under development using Unity 3D game engine. Furthermore, we plan to analyze the quality of social interactions during a collaborative game session in order to more deeply understand the dynamics of affective states over time.

6. ACKNOWLEDGMENTS

The authors would like to thank EVERSIM (Simulation & Serious Games) society for providing them the iScen platform and assisting them in its use.

7. REFERENCES

- [1] Aghababvan, A. 2014. E3: Emotions, Engagement and Educational Games. *International Educational Data Mining Society*.
- [2] Daoudi I., Chebil R. and Lejouad Chaari W. 2018. A Novel Tool to Predict the Impact of Adopting a Serious Game on a Learning Process. In *Proceedings of the 20th International Conference on Enterprise Information Systems*. 1:585-592.
- [3] Walker, W. E., Giddings, J., and Armstrong, S. 2011. Training and learning for crisis management using a virtual simulation/gaming environment. *Cognition, Technology & Work*, 13(3), 163-173.
- [4] Daoudi I., Chebil R., Tranvouez E., Lejouad Chaari W., and Espinasse B. 2017. Towards a Grid for Characterizing and Evaluating Crisis Management Serious Games: A Survey of the Current State of Art. *International Journal of Information Systems for Crisis Response and Management (IJISCRAM)*, 9, Issue 3, 76-95.
- [5] A. Haferkamp N., and Kraemer N. C. Linehan. C., Schembri, M. 2011. Training disaster communication by means of serious games in virtual environments. *Entertainment Computing*, 2(2), 81-88.
- [6] Mendez, G., Avramides, K., de Freitas, S., and Memarzia, K. 2009. Societal impact of a Serious Game on raising public awareness: the case of FloodSim. In *Proceedings of the ACM SIGGRAPH Symposium on Video Games*, 15-22.
- [7] Oulhaci M.A., Tranvouez E., Fournier S., and Espinasse B. 2015. Improving Players' Assessment in Crisis Management Serious Games: The SIMFOR Project. In *Information Systems for Crisis Response and Management in Mediterranean Countries*, 85-99.
- [8] Taillandier F., and Adam C. 2018. Games Ready to Use: A Serious Game for Teaching Natural Risk Management. *Simulation & Gaming*, 49, 441-470.
- [9] Silva, V., Dargains, A., Felício, S., and Carvalho, P. and al. 2014. Stop disasters: serious games with elementary school students in Rio de Janeiro. In *8th International Technology, Education and Development Conference*, 1648-1659.
- [10] Theo van, R., Igor, M., and Mark de, B. 2015. Multidisciplinary coordination of on-scene command teams in virtual emergency exercises. *International Journal of Critical Infrastructure Protection*, 9, 13-23.
- [11] Shute, V. J. 2011. Stealth assessment in computer-based games to support learning. *Computer games and instruction*, 55(2), 503-524.
- [12] Guthier B., Dörner R., and Martinez H.P. 2016. Affective Computing in Games. *Entertainment Computing and Serious Games*, 9970, 402-441.
- [13] <https://indico.io/blog/docs/indico-api/text-analysis/>
- [14] Tadas B., Amir Z., Yao C.L., and Louis-Philippe M. 2018. OpenFace 2.0: Facial Behavior Analysis Toolkit. *13th IEEE International Conference on Automatic Face & Gesture Recognition*.
- [15] Ekman, P. 1999. Basic emotions. In *T. Dalgleish & M. J. Power (Eds.), Handbook of cognition and emotion*, 45-60. New York, NY, US: John Wiley & Sons Ltd.
- [16] Friesen, W., and Ekman, P. 1983. EMFACS-7: Emotional Facial Action Coding System. *Unpublished manual, University of California*. <https://www.paulekman.com/>
- [17] Soujanya P., Erik C., Rajiv B., and Amir H. 2017. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37, 98-125.
- [18] Tan C. T., Rosser D., Bakkes S., and Pisan Y. 2012. A Feasibility Study in Using Facial Expressions Analysis to Evaluate Player Experiences. In *Proceedings of The 8th Australasian Conference on Interactive Entertainment: Playing the System*, New York, NY, USA.
- [19] Ramin T., Ashish A., Troy M., and Sethuraman P. 2018. Real-time stealth intervention for motor learning using player flow-state. *IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH)*.
- [20] Kuncheva L. I. 2002. A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2), 281 –286.
- [21] Aigrain, J., Spodenkiewicz, M., Dubuisson, S., Detyniecki, M., Cohen, D., and Chetouani, M. 2016. Multimodal stress detection from multiple assessments. *IEEE Transactions on Affective Computing*.
- [22] Alberdi, A., Aztiria, A., and Basarab, A. 2016. Towards an automatic early stress recognition system for office environments based on multimodal measurements: A review. *Journal of biomedical informatics*, 59, 49-75.
- [23] Gao, H., Yüce, A., and Thiran, J. P. 2014. Detecting emotional stress from facial expressions for driving safety. In *IEEE International Conference on Image Processing (ICIP)*, 5961-5965.
- [24] D'Mello, S., and Graesser, A. 2012. Dynamics of affective states during complex learning. *Learning and Instruction*, 22(2), 145-157.
- [25] Craig S., D'Mello S., Johnson A., and Graesser A. 2008. Emote aloud during learning with AutoTutor: Applying the Facial Action Coding System to cognitive-affective states during learning. *Cognition and Emotion*, 22 (5), 777-788.
- [26] Csikszentmihalyi M. 1990. Flow: The psychology of optimal experience. *Harper & Row*.
- [27] Michalis, F. 2016. A Review of Emotion-Aware Systems for e-Learning in Virtual Environments. *Chapter11, Formative Assessment, Learning Data Analytics and Gamification. In ICT Education Intelligent Data-Centric Systems*, 217-242.
- [28] Jeanne H.B., Christine M.F., Kathleen A.C., Evan M. , Kimberly M.B., and Jacquelyn N.P. 2009. The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing. *Journal of Experimental Social Psychology*, 45, Issue 4, 624-634.
- [29] <http://www.i-scen.com/home.php?lang=en>
- [30] Daoudi I., Tranvouez E., Chebil R., Espinasse B., and Lejouad Chaari W. 2017. Learners' Assessment and Evaluation in Serious Games: Approaches and Techniques Review. In: Dokas I., Bellamine-Ben Saoud N., Dugdale J., Díaz P. (eds) *Information Systems for Crisis Response and Management in Mediterranean Countries. ISCRAM-med 2017. Lecture Notes in Business Information Processing*, vol 301. Springer, Cham.

Exploration of Process Mining Opportunities In Educational Software Engineering - The GitLab Analyser

Philipp Dumbach, M. Sc.¹
philipp.dumbach@fau.de

Alexander Aly, B. Sc.¹
alexander.aly@fau.de

Markus Zrenner, M. Sc.¹
markus.zrenner@fau.de

Prof. Dr. Bjoern M. Eskofier, PhD
¹Machine Learning and Data Analytics Lab
Department of Computer Science
Friedrich-Alexander-Universität Erlangen-Nürnberg
Carl-Thiersch-Str. 2b, 91052 Erlangen
bjoern.eskofier@fau.de

ABSTRACT

The increasing complexity in software development leads to the necessity for a detailed data analysis. Literature illustrates a stronger research focus on Educational Process Mining (EPM) being applied to the fields of e-learning and professional training. In this work, the opportunities of Process Mining (PM) are further examined by the evaluation of software engineering (SE) courses. The methodology follows the five stages of the *L* life cycle model* for PM projects using data from software repositories. The event log data was analyzed with the PM tool Disco to examine the students' work following an agile development process. The new tool *GitLab Analyser* supports supervisors to visualize educational processes and still extracts event logs for the further analysis and application of PM techniques.

Keywords

data mining, educational process mining, software engineering, agile development, Git, GitLab, education, software repositories, Innovation Lab, Scrum

1. INTRODUCTION

Within the last decades an enormous increase of research interest associated to the field of machine learning systems was observed [7]. Especially during the last ten years, the public interest in the impact of applied machine learning and data analysis methods further grew [6]. The massive increase and demand of new software functionality in these fields also lead to higher software complexities. Dealing with these complexities is difficult especially when it comes to innovations. For the development process of innovative software, time-to-market is a relevant factor due to its impact on revenue and business success of companies in comparison to their potential competitors [9].

In order to cope with the raised complexity Version Control Systems (VCS) were deployed in software development. Ad-Philipp Dumbach, Alexander Aly, Markus Zrenner and Bjoern M. Eskofier "Exploration of Process Mining Opportunities In Educational Software Engineering - The GitLab Analyser" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 601 - 604

ditionally, time-boxed plans, IT systems like content management systems as well as issue trackers are now widely used [1, 9].

Those systems also become more and more important for the educational domain. In practical SE courses, students learn and use such systems in order to better structure their development process as well as become prepared for their future employments. Research picked up on this development and started to explore PM opportunities regarding the evaluation of educational software development teams in order to improve the learning process. By extracting event logs from the software development projects, critical processes can be identified and improved.

Tools for Educational Process Mining:

Different tools for extracting, visualizing and analyzing event logs for educational purposes were introduced in literature. One solution called *SoftLearn* is mentioned in the work of Vázquez-Barreros et al. [3] and allows the visualization of the students' learning paths by offering a graphical user interface (GUI).

A publicly available solution is the platform *PHIDIAS* presented by Awatef et al. [2]. This tool provides a service for data and process mining to educational experts. It supports the reconstruction of educational processes and the detailed analysis of social networks.

Sokol et al. [11] introduced a web application called *MetricMiner* for mining software repositories and supporting researchers with the data extraction and statistical inference. Another analysis tool in the application area of Git repositories is *Gitinspector*. This tool is not directly defined as a PM tool, but supports creating insight into development processes by analyzing Git logs and delivering details about the author's contribution over time [5].

Despite all those approaches, Bogarin et al. [4] underline the lack of tools supporting educational specialists from various fields in analyzing educational processes by providing an easy to use tool and a generic framework for EPM in the context of SE courses. Many of the tools demand special knowledge in fields which educational specialists lack.

Applications of Educational Process Mining:

Bogarin et al. [4] summarized various application domains of EPM, which are listed in Table 1.

Table 1: Application areas for EPM [4]

Application field	Amount of studies
Massive Open Online Courses, hypermedia learning environments, learning management systems	8
Computer-supported collaborative learning	5
Professional Training	5
Curriculum Mining	3
Computer-based assessment	2
Software repositories	2

In these applications EPM is used to discover learning flows and sequential patterns. Participants’ decision-making processes as well as usage of group communication tools are analyzed to detect learning difficulties. Consequently, the quality of education can be improved by adapting the educational software development process based on the analysis results [4].

Mittal et al. [8] introduced a holistic approach to evaluate the complete educational software development process. They present the idea for a research framework for PM using event logs of VCSs, issue tracking systems and team wikis. To the best of our knowledge, there is no tool available extracting all the relevant event logs necessary to feed this research framework.

Contribution

We contribute a tool called *GitLab Analyser*, which visualizes and extracts EPM relevant event logs from the open source software project management framework GitLab. The tool is easy to use not only for experts but also general educational specialists. Besides, it allows a holistic analysis over underlying learning processes by extracting event logs from the git software repository, the GitLab issue tracker and the GitLab documentation Wiki.

The *GitLab Analyser* is publicly available as standalone application under the following link:

<https://www.mad.tf.fau.de/research/gitlab-analyser/>.

2. METHODOLOGY:

For the development of the tool we aligned with the first three stages in the five stage process of the *L* life-cycle model* for PM projects as described in the PM manifesto: planning and justification, data inspection, event log extraction, analysis execution and result interpretation [12, 13].

Planning and justification: We planned to extract event logs from a SE course called *Innovation Lab for Wearable and Ubiquitous Computing* offered by the Machine Learning and Data Analytics Lab at the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) over the last five semesters. The Innovation Lab is offered to various majors of the university’s technical faculty. In this course interdisciplinary student groups of size five to eight develop innovative software related prototypes in cooperation with multiple industry partners and public institutions. Over four months, these teams use the agile development process Scrum [10] and perform three Sprints after an on-boarding phase.

Data inspection: As a project management tool, GitLab

Community Version 12.8.0 was used because it offered

- planning features (milestones and issue tracker),
- versioning of the source code and
- documentation features (Wiki).

Both the VCS of the source code as well as the Wiki are git repositories. From all those features events describing the development process can be extracted.

Event log extraction: The events of the VCS and the Wiki were extracted using the git native ‘git log’ command. Events related to the planning features (milestones, issues) were extracted using GitLab’s native API and the REST API client postman. After extracting the events, they were converted to a .CSV file, which can be interpreted by commonly used PM software like Disco or Celonis.

Table 2 summarizes the data set out of the Innovation Lab projects at FAU, the tool was developed with.

Table 2: FAU GitLab log data

GitLab General Information	Value
Number of projects	24
GitLab issues	3409
GitLab repositories commits	5332
GitLab wiki commits	8474
Number of project branches	744

3. RESULTS AND DISCUSSION

3.1 Event Logs for Process Mining

Table 3 gives an overview about the majority of events and activities concerning the planning in GitLab, tracking of source code changes in Git as well as the documentation of the project in Wiki.

Table 3: Events and activities considered in project development process

Planning	issues, issue labels, milestones, branches, merge requests, notes, projects
VCS	number of changed files, commits, commit type, inserted and deleted lines of code, days with commits, number of merges and merge requests
Documentation	inserted and deleted lines of code, number of wiki pages, commits, days with commits

Further indirect available events are extracted by mining the issue notes section e.g. *changed milestone*, *assigned to* or *time spent*. By extracting the data the minimum information about the event logs is collected (*instance id*, *activity*, *timestamp*, *actor*).

3.2 The GitLab Analyser

The *GitLab Analyser* is developed for the implementation as easy to use tool for general educational specialists and to visualize event logs for supervisors. The tool offers three different analysis types all aiming to support supervisors in evaluating students and the development process itself:

1. *Single project analysis:* Support for supervisors in evaluating students and the development process itself.
2. *Group project analysis:* Support for supervisors to compare the performance of different teams of the same course (given that all projects are hosted on the same GitLab server).
3. *Cross-project analysis:* Support for different courses to compare the development process by extracting event logs from different GitLab servers.

Within the tool different result views are presented to the user on a dashboard. A user view offers the analysis of project events performed by the individual users, whereas a project view provides insights into the overall project status.

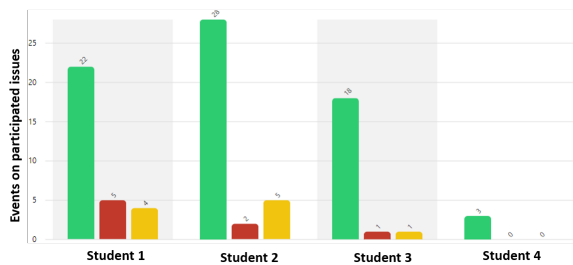


Figure 1: Participation event distribution of users. Color coding: assigned to user (green); unassigned to user but still participated (red) and mentioned user and participated (yellow).

Figure 1 depicts an exemplary graph from the dashboard on the user view. This graph visualizes the number of events as a result of the issues students worked on during the development process and whether they were assigned to those issues. This example shows an even assignment rate with one exception, *Student 4*. Based on this visualization the supervisors can see students with less participated issues and act based on these results by providing additional help to the student in case of lack of background knowledge, breaking big issues down into smaller issues to increase the student's success or motivation.

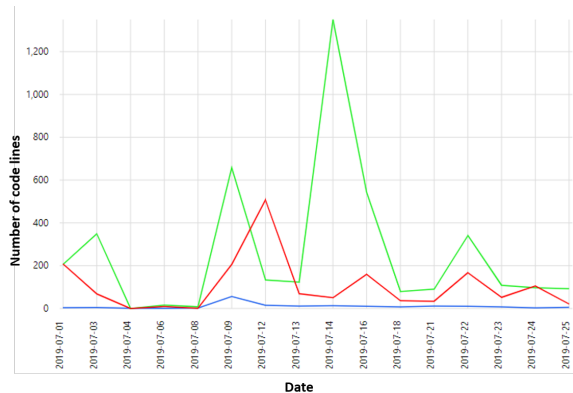


Figure 2: Inserted (green) and deleted (red) code lines in commits and changed files (blue) per day.

Figure 2 shows a graph from the project view. This graph visualizes the number of added, deleted and edited software code lines committed to the VCS of all team members over time. It clearly shows a peak in the middle of the development process, which was due to a Scrum Review at this point in time. By inspecting this graph, supervisors can identify that students do not continuously push their code changes to the repository, which is necessary for other team members to work on a common base. Thus, they can motivate students to improve the development process by continuously committing their new developments to GitLab.

3.3 PM opportunities in university projects

With Celonis and Disco two PM tools were tested. The extracted and transformed event log data (from GitLab raw-log data) was exported and analyzed with Disco to support the identification of correlations in the development process. In the Wiki and Git analysis the commit behaviour and distribution of commit activities was identified. In addition to the visualization of issue states, performance measurements like the average working time on an issue or time until the first user assignment, were determined by the analysis of GitLab features. The participation on issues as well as the information about users carrying out an activity at a specific point in time can be visualized. The time-boxes filter options in Disco enable to use the event logs for precise analysis of activities occurring for example within one Sprint. Furthermore, Disco offered options to analyze specific process parts by filtering the individual and process-relevant activities.

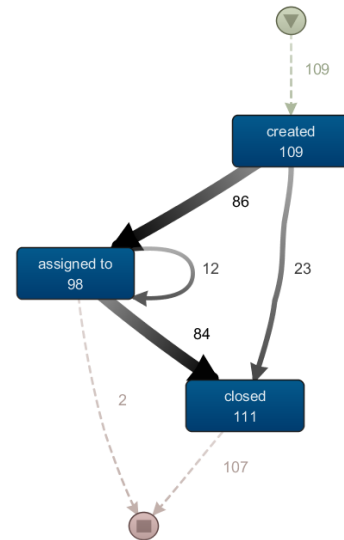


Figure 3: Process Map - Status changes of issues (100 percent activities, 50 percent paths)

Figure 3 illustrates the analysis of assignment activities in form of a process map as an exemplary Disco visualization. The investigation shows the total number of issues being assigned before their closing, i.e. whether a team member was responsible for them. Additionally, the number of assignee changes can be viewed, represented by the self-referencing arrow at the "assigned to" activity (12 times), and used as indicator for evaluating the team performance.

The event log extraction allowed to gain insight into the

students' work following an agile development process. Nevertheless, the analysis and filter configurations in Disco require a certain training period. It was critically questioned whether scientific staff, performing the role as Scrum Master not as a data scientist, need to familiarize themselves with the deeper functionality of PM analysis. Instead, course supervisors should be enabled with a tool to obtain essential analysis results with less effort in a short amount of time.

4. SUMMARY AND OUTLOOK

Due to the increasing complexity in the software development process the application of PM techniques offers valuable opportunities especially in the education domain. Various studies underlined the necessity for tools supporting educational analysis following an agile development process [4]. We introduced a standalone, easy to use tool called *GitLab Analyser* which can be used by supervisors from various fields without significant background in computer science. The tool not only offers the event log extraction for a detailed PM analysis using elaborate PM software (e.g. Disco, Celonis), but also visualizes the individual event logs in clear way for supervisors to evaluate students and the development process quickly. We made the tool publicly available under the following link:

<https://www.mad.tf.fau.de/research/gitlab-analyser/>.

The *GitLab Analyser* will be used within the upcoming semester of the Innovation Lab by the supervisors of the different development teams for immediate feedback on the development process. Additionally, the first version will be available for supervisors of other universities with similar courses to receive feedback and first bug reports for the next iteration of the tool development process.

Besides, we will use the tool to extract event logs from the last five semesters of the FAU's Innovation Lab and other comparable innovation courses of cooperating universities. By finishing the *L* life-cycle model* for PM projects through performing analysis execution and result interpretation, we will evaluate students' development and learning processes in order to come up with recommendations for improved teaching.

5. ACKNOWLEDGMENTS

The authors would like to thank the Zentrum.Digitalisierung Bayern (ZD.B) for funding the *Innovation Lab for Wearable and Ubiquitous Computing*. We thank Prof. Dr. Gerd Beneken and Martin Kucich from the Rosenheim Technical University of Applied Sciences as well as Prof. Dr. Daniela Nicklas and Simon Steuer from the University of Bamberg for sharing their expertise regarding the conduction of SE courses and supporting the increase of project data for the upcoming steps.

Bjoern M. Eskofier gratefully acknowledges the support of the German Research Foundation (DFG) within the framework of the Heisenberg professorship programme (grant number ES 434/8-1).

6. REFERENCES

- [1] H. Awatef, B. GUENI, M. Fhima, A. CAIRNS, and S. David. Process mining in the education domain. *International Journal on Advances in Intelligent Systems*, volume 8 no 1&2:pages 219–232, 2015.
- [2] H. Awatef, B. GUENI, M. Fhima, A. CAIRNS, S. David, and N. KHELIFA. Towards custom-designed professional training contents and curriculums through educational process mining: Process mining in the education domain. *IMMM 2014 : The Fourth International Conference on Advances in Information Mining and Management*, 2014.
- [3] B. V. Barreiros, M. Lama, M. Mucientes, and J. C. Vidal. Softlearn: A process mining platform for the discovery of learning paths. In *2014 IEEE 14th International Conference on Advanced Learning Technologies*, pages 373–375, 2014.
- [4] A. Bogarín, R. Cerezo, and C. Romero. A survey on educational process mining. *WIREs Data Mining and Knowledge Discovery*, 8(1):e1230, 2018.
- [5] Gitinspector. <https://github.com/ejwa/gitinspector>: Accessed on march 9, 2020., 2012.
- [6] Iain M. Cockburn, Rebecca Henderson, and Scott Stern. The impact of artificial intelligence on innovation: An exploratory analysis. In Ajay Agrawal, Joshua Gans, and Avi Goldfarb, editors, *The Economics of Artificial Intelligence: An Agenda*, pages 115–146. University of Chicago Press, 2019.
- [7] Jiaying Liu, J. Liu, X. Kong, F. Xia, X. Bai, L. Wang, Q. Qing, and I. Lee. Artificial intelligence in the 21st century. volume 6, pages 34403–34421.
- [8] M. Mittal and A. Sureka. Process mining software repositories from student projects in an undergraduate software engineering course. In *Companion Proceedings of the 36th International Conference on Software Engineering, ICSE Companion 2014*, pages 344–353, New York, NY, USA, 2014. Association for Computing Machinery.
- [9] N. M. Devadiga. Software engineering education: Converging with the startup industry. In *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE T)*, pages 192–196, 2017.
- [10] K. Schwaber. Scrum development process. In J. Sutherland, C. Casanave, J. Miller, P. Patel, and G. Hollowell, editors, *Business Object Design and Implementation*, pages 117–134, London, 1997. Springer London.
- [11] F. Sokol, M. Aniche, and M. A. Gerosa. Metricminer: Supporting researchers in mining software repositories. pages 142–146, 2013.
- [12] W. van der Aalst, A. Adriansyah, de Medeiros, Ana Karla Alves, M. Westergaard, and M. Wynn. Process mining manifesto. In F. Daniel, K. Barkaoui, and S. Dustdar, editors, *Business Process Management Workshops*, pages 169–194, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [13] M. L. van Eck, X. Lu, S. J. J. Leemans, and W. van der Aalst. Pm²: A process mining project methodology. In J. Zdravkovic, M. Kirikova, and P. Johannesson, editors, *Advanced Information Systems Engineering*, pages 297–313, Cham, 2015. Springer International Publishing.

Comparing and Combining Tests for Plagiarism Detection in Online Exams

Edward F. Gehringer, Xiaohan Liu, Abhirav Dilip Kariya, and Guoyi Wang
North Carolina State University
{efg, xliu74, akariya, gwang25}@ncsu.edu

ABSTRACT

Online exams with machine-readable answers open new possibilities for plagiarism and plagiarism detection. Each student's responses can be compared with all others to look for suspicious similarities. Past work has developed several approaches to detecting cheating: n -gram similarity, Levenshtein distance, Smith-Waterman distance, and binomial probability. To that we add our own term-frequency based approach, called the "weirdness vector," which measures how unusual a student's answers are, compared to all other students. Each of these approaches seems suited to particular question types. Levenshtein and Smith-Waterman are suited to long text strings, as appear in answers to essay questions. Binomial probability and n -gram similarity are well suited for finding suspicious patterns in responses to multiple-choice questions. The "weirdness vector" is most applicable to fill-in-the-blank questions.

Unlike past research, that applied a single metric to detect cheating in an exam with questions of a single type, this paper measures how different approaches work with different kinds of questions, and proposes methodologies for combining the approaches for exams that consist of all three kinds of questions. This work shows promise for detecting cheating in open-web exams, where students can cheat using covert Internet channels, and is especially applicable in situations where exams cannot be proctored.

Keywords

Online exams; plagiarism; Levenshtein distance; n -grams

1. INTRODUCTION

Online exams have become more common in recent years due to the growth in online courses, especially after the transition to emergency online instruction. They have the advantage of faster grading, especially for distance ed, more copious feedback, and they can provide a more authentic testing environment by allowing students to access certain

information from the web (e.g., the course notes) during the exam.

Yet open-web exams do raise concerns about cheating [1]. Browsers can be locked down, and students can be monitored remotely with cameras [2]. But monitoring is expensive, and locking down browsers may destroy the authenticity of the environment. For example, in a course on open-source coding, students would always do their work online. If they don't have access to the Internet during an exam, they must work in an environment far different from their usual one. However, an authentic testing environment can only be used if there is a way to detect plagiarism.

Our approach is to use data mining to measure the similarity of the submitted answers. We extend our past work [3] by incorporating additional published tests into our application, and studying their applicability to different types of questions. Section 2 covers tests that have been proposed by others. Section 3 introduces new techniques for handling particular kinds of questions. Section 4 reports our findings from experiments on real data, and discusses which metrics are suitable for which types of questions. Section 5 summarizes our work and points out ideas for future progress.

2. RELATED WORK

Many published papers address automated detection of plagiarism, but with few exceptions, each paper focuses on a single mathematical test. While a few papers [4] do consider multiple tests, they do so in the context of comparing competing tests for detecting plagiarism on a particular kind of question (e.g., multiple choice). Since exams contain many different kinds of questions (multiple choice, essay, fill in the blank, matching, etc.) what is needed is a single application that can apply appropriate tests to responses to different kinds of questions. That is the goal of our research.

2.1 Levenshtein Distance

The Levenshtein distance between two strings is the minimum number of edits required to change one string into the other. For example, the Levenshtein distance between "faculty" and "faulty" is 1, the Levenshtein distance between "sloop" and "sleep" is 2, and the Levenshtein distance between "country" and "countries" is 3. In the research of investigating whether a machine learning model based on a statistical method works better than a model based on a structural method, the Levenshtein distance was chosen to be the similarity measurement for the structural approach.

Edward Gehringer, Xiaohan Liu, Abhirav Kariya and Guoyi Wang "Comparing and combining tests for plagiarism detection in online exams" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 605 - 609

Levenshtein distance has been researched not only for traditional string match, but as a structural method in clustering-based machine learning models of plagiarism detection [5].

One limitation of Levenshtein distance in detecting plagiarism is that rearrangement of text produces a large Levenshtein distance, since Levenshtein distance is focused on one-character (or one-word) edits. Suppose that two students' answers, taken as a whole, bear little resemblance to each other, but they contain sequences in different positions that are highly similar. The Smith-Waterman algorithm can identify this.

2.2 Smith-Waterman Algorithm

The Smith-Waterman algorithm is another classical string similarity metric. It looks for similar local regions to identify optimal sequence alignments. For example, the best alignment of two sequences $X = \text{"abcbadb"}$ and $Y = \text{"abdbd"}$ would be

```
a b c b a d b
a b - b - d b
```

Researchers have proposed alterations of the Smith-Waterman algorithm that were tested effective in practice of detecting collusion while speeding up the algorithm without using up much space [6]. Traditional Smith-Waterman algorithm searches through a pair of sequences and finds the maximum piece of consecutive matching characters, whereas the revised implementation introduces the cut-off concept to keep track of multiple pieces of matching. The modification yields more optimal local alignments and thus more effective on plagiarism detection as well.

2.3 n -grams

Another attempt from the structural perspective is n -grams. We can consider a word as a token [7]. Then an n -gram is a set of n consecutive words. Then for two exam submissions, we can ask what is the longest common n -gram between them, or how many n -grams of length $> k$ do they have in common? This is a useful metric for comparing two students' essay answers, but it also useful for comparing other kinds of answers, such as answers to multiple-choice (MC) questions. Here, MC answers, not words, make up the strings we are comparing.

MC questions have the property that the answers are chosen from a discrete set, usually about four in cardinality. Given that there are m possible answers for each question, the probability that two students will choose the same answer by chance is $\frac{1}{m}$. The probability that they will choose the same k consecutive answers is $\frac{1}{m^k}$. This is the idea behind the binomial test [8]; it is very unlikely that two students will choose a large number of the same wrong answers by chance.

Each of these methods works well on a specific type of text. A more comprehensive approach that works on all types of questions is needed for online exams. We will further analyze the effectiveness of each metric to determine which metrics work better for multiple-choice questions, fill-in-the-blank questions, and essay questions, respectively.

3. PROPOSED METHODS

3.1 The "weirdness" vector

The weirdness-vector metric looks for pairs of students who have similar but unusual answers. The basic idea is to calculate the term frequency of each response by each student and create a vector of term frequencies. Then we can use cosine similarity to measure the distance between the weirdness vectors of each pair of students. Those who have the most similar vectors are worth further inspecting.

3.1.1 Data Preprocessing

1. For the set of students $S = s_1, s_2, \dots, s_n$, we extract all their responses R into a matrix where $r_{i,j}$ is the response to question q_i by student s_j .
2. Then we remove the stop words and punctuation in the response matrix.
3. We use a function to classify each question on the exam as belonging to one of three question types: Multiple-choice, fill-in-the-blank, and free-response essay questions.

3.1.2 Implementation

1. For each response $r_{i,j}$ of student s_j to question q_i , we calculate its term frequency among all the responses to question q_i . Each response $r_{i,j}$ is converted into a "bag of words," and is compared with every other bag-of-words response to question q_i . The number of occurrences of each bag of words divided by the number of students n gives us the frequency $f_{i,j}$ of a response $r_{i,j}$.

$$f_{i,j} = \frac{\text{number of times } r_{i,j} \text{ appears in responses to } q_i}{n}$$

2. It is the low term frequencies that may be suspicious, but for the other tests in the program, high values are suspicious. Hence, we calculate the inverse term frequency instead:

$$w_{i,j} = 1 - f_{i,j}$$

3. Each student s_j has a "weirdness" vector W_j consisting of the inverse frequencies $w_{i,j}$ of each response to each question q_i , i.e., $W_j = w_{1,j}, w_{2,j}, \dots, w_{m,j}$, where q_1, q_2, \dots, q_m are the questions in the test.
4. We use cosine similarity to measure the closeness between pairs of weirdness vectors. For a pair of vector X and Y , the cosine similarity is calculated as

$$\text{cosine similarity} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

where x_i and y_i , $i = 1, 2, \dots, n$ are components of X and Y . The vectors with similarly high inverse frequencies will return high cosine similarity, for small values in the vector components do not contribute much when calculating the cosine similarity. That being said, only similar but unusual responses will stand out in similarity scores.

3.1.3 Regarding identical answers

The weirdness vector highlights suspicious behavior by detecting pairs of exams that contain identical incorrect answers, yet it is also worthwhile to identify pairs of exams that have identical correct answers. Some questions have multiple possible answers, where students can answer correctly without necessarily providing the exact same responses. Such cases can be well addressed by an algorithm that takes multiple correct answers into account; however, no algorithm can detect plagiarism among students who have provided the same correct response where only one correct response is possible.

3.2 Bag-of-Words Extension

Several metrics help detect plagiarism in text-based answers. Results can be enhanced by preprocessing the text before applying metrics. One kind of preprocessing is getting rid of stop words and removing punctuation. We can go one step further and treat the remaining words as an unordered set. This is the bag-of-words model.

3.2.1 Use Case

To illustrate the advantage of the bag-of-words model for finding similar answers, consider this example:

```
Response1 = "pattern: strategy"
Response2 = "strategy pattern"
Response1 == Response2 // False
bag_of_words(remove_stop_words(Response1)) ==
bag_of_words(remove_stop_words(Response2)) // True
```

Given that these responses are deemed incorrect, it is worthwhile to count the two wrong answers as matching. Without the removal of stop words and bag-of-words analysis, this case would go unnoticed as evidence of potential plagiarism.

4. EMPIRICAL RESULTS

The research questions that we are trying to answer are whether the tests can detect suspicious similarity, as well as which tests are most effective on each type of questions. We consider a test effective if it produces only a few unusual values (outliers) among its results. Of course, results from tests alone cannot be solid evidence of cheating; instructors would need to inspect the exam papers. To forestall excessive manual inspection, a good test should direct attention to the few most suspicious responses. If the observed values given by a metric contain outliers when it is applied to a particular kind of question, this metric can be deemed useful for that type of question.

We can use data visualizations to illustrate the effectiveness of all the metrics on three types of questions. We used real exam data from CSC 517 (all offerings between Fall 2014 and Spring 2020) at North Carolina State University. All data was de-identified before use.

4.1 Effectiveness of each metric

The weirdness metric shows us (Figure 1) how unusual it is for a pair of exams to share the same wrong answer to a question. Weirdness is a good test for FiB exams if only a few exams have a large number of the same unusual incorrect

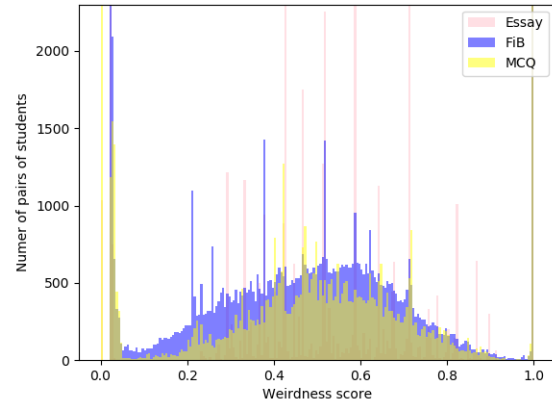


Figure 1: Weirdness metric on 3 types of questions

answers. On essay questions, however, it is the responses with high term frequency that are suspicious, since each response should have its unique phrasing. Essentially, each incorrect essay response is considered “weird” and hence, the weirdness values will show a discrete distribution as shown in the histogram above. For MC questions, there is a much smaller number of possible choices, and thus, weirdness does not work as effectively as for FiB. Though both the FiB and MC weirdness values have small tails, the values are more meaningful for FiB questions.

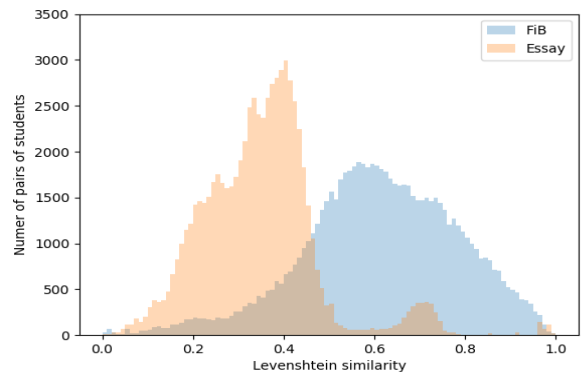


Figure 2: Levenshtein on FiB and essay questions

The Levenshtein metric (Figure 2) uses edit distance to compute string similarity. While weirdness watches for short unusual responses for FiB questions, Levenshtein has its strength in detecting long similar responses for essay questions. As the histogram for Levenshtein performance shows, Levenshtein generates many fewer outliers on essay questions than on FiB questions. The essay histogram has a minor peak near 1.0, highlighting the responses that are suspiciously similar, whereas the FiB histogram has many high values, suggestive of false positives.

Smith-Waterman is pretty good at comparing long texts, and it is much more revealing on essay questions than on

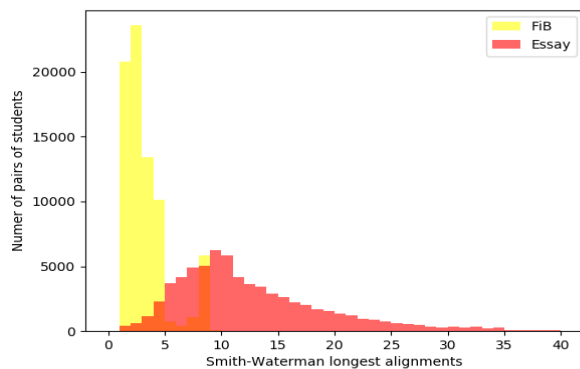


Figure 3: Smith-Waterman metric results

FiB questions, as we can tell from the graph above.

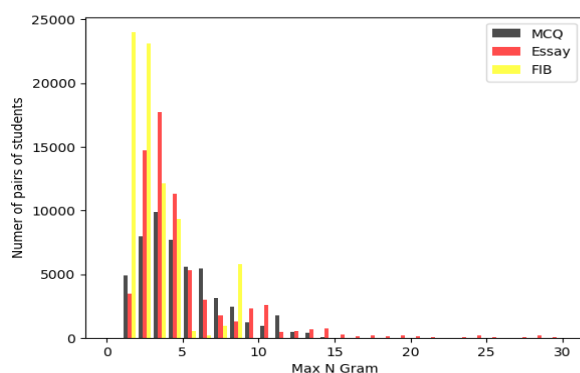


Figure 4: Max n -grams metric results

n -grams are naturally suited to finding long similarities in essay questions. We can also concatenate all the MC responses of two students and use n -grams to compute the longest common subsequence between them. Since the number of pairs drops significantly after max n -gram length = 10 for essay questions, we choose 10 as the threshold and consider those greater than 10 to be outliers. FiB responses are much shorter, typically no longer than 7 words, and are expected to be mostly identical; thus n -grams are unlikely to provide much guidance. The max n -grams lengths of MC responses tell us how many consecutive MC questions two students answered identically.

The n -gram metric can, of course, help detect students who were collaborating extensively on MC questions, but it does not take correctness of the responses into account. Consecutive same correct MC responses should not be treated as suspicious.

Binomial is used for MC questions only, as it calculates the probability of students having the same wrong answers. It is a more reasonable metric for MC questions than n -grams because it does take correctness of responses into account.

4.2 The most suitable metric

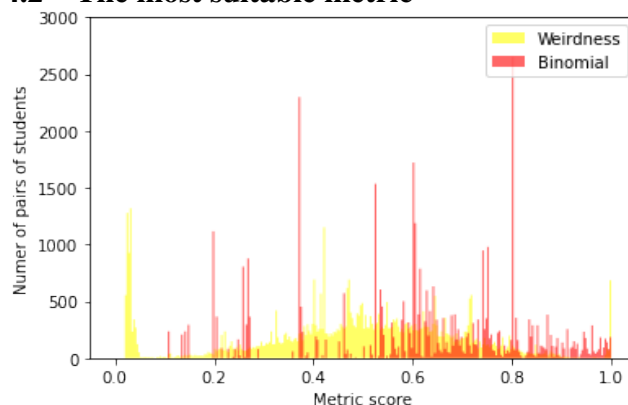


Figure 5: Different metrics on MC questions

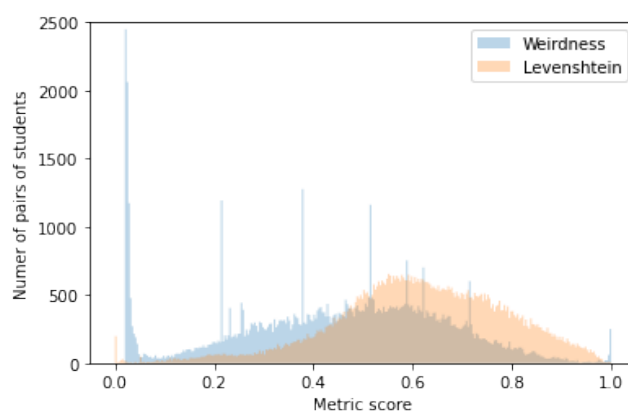


Figure 6: Different metrics on FiB questions

As discussed earlier, weirdness is much more applicable to FiB questions than other string matching metrics. The concave upward curve at (0.8, 1.0) justifies the effectiveness of weirdness.

Levenshtein, Smith-Waterman, and N -grams are all good metrics for essay questions, although empirically, Levenshtein is more effective over other tests.

5. SUMMARY

We can conclude from the empirical results that for multiple-choice questions, one should seek help from the binomial test. For fill-in-the-blank questions, weirdness works the best. For essay questions, Levenshtein, Smith-Waterman, and n -grams all work effectively.

6. REFERENCES

- [1] G. Fenu, M. Marras, and L. Boratto, "A multi-biometric system for continuous student authentication in e-learning platforms," *Pattern Recognition Letters*, vol. 113, pp. 83–92, 2018.
- [2] Y. Atoum, L. Chen, A. X. Liu, S. D. Hsu, and X. Liu, "Automated online exam proctoring," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1609–1624, 2017.

- [3] S. Biswas, D. G. Edward F. Gehringer, S. Sahane, , and S. Sharma, "A data-mining approach to detecting plagiarism in online exams," in *EDM 2018, Proceedings of the 11th International Conference on Educational Data Mining*, pp. 504–507.
- [4] C. Zopluoglu, "Similarity, answer copying, and aberrance: Understanding the status quo," *Handbook of quantitative methods for detecting cheating on tests*, pp. 25–46, 2017.
- [5] E. Anzén, "The viability of machine learning models based on levenstein distance and cosine similarity for plagiarism detection in digital exams," 2018.
- [6] R. W. Irving, "Plagiarism and collusion detection using the smith-waterman algorithm," *University of Glasgow*, vol. 9, 2004.
- [7] M. Zini, M. Fabbri, M. Moneglia, and A. Panunzi, "Plagiarism detection through multilevel text comparison," in *2006 Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS'06)*, pp. 181–185, IEEE, 2006.
- [8] F. S. Belleza and S. F. Belleza, "Detection of cheating on multiple-choice tests by using error-similarity analysis," *Teaching of Psychology*, vol. 16, no. 3, pp. 151–155, 1989.

Curriculum profile: modelling the gaps between curriculum and the job market

Aleksandr Gromov
University of Technology
Sydney
PO Box 123, Broadway,
Sydney, NSW, 2007
aleksandr.gromov
@uts.edu.au

Andrei Maslennikov
University of Technology
Sydney
PO Box 123, Broadway,
Sydney, NSW, 2007
andrei.maslennikov
@uts.edu.au

Nikolas Dawson
University of Technology
Sydney
PO Box 123, Broadway,
Sydney, NSW, 2007
nikolas.dawson@uts.edu.au

Katarzyna Musial
University of Technology
Sydney
PO Box 123, Broadway,
Sydney, NSW, 2007
katarzyna.musial-
gabrys@uts.edu.au

Kirsty Kitto
University of Technology
Sydney
PO Box 123, Broadway,
Sydney, NSW, 2007
kirsty.kitto@uts.edu.au

ABSTRACT

This study uses skill-based curriculum analytics to mine the curriculum of an entire university. A curriculum profile is constructed, providing insights about university curriculum design and the match between one institution's curriculum and the job market for a cluster of data-intensive fields. Automating the delivery of diagnostic information like this would enable institutions to ensure that their professionally-oriented degrees meet the needs of industry, so helping to improve learner outcomes and graduate employability.

Keywords

curriculum mapping, curriculum profile, skill-based curriculum analytics, ontologies, skills profile, job market

1. INTRODUCTION

People around the world see universities as an important step in building a successful career [3]. They invest time and finances in undergraduate and postgraduate courses, with a goal to gain new competencies that will help them to find a job [2]. Over the decades, a number of institutions in the tertiary sector have worked hard to adapt their curriculum to market requirements, seeking to prepare more work-ready graduates. However, there is an ongoing debate about whether university efforts to develop students' skills have a noticeable influence upon graduate employability [18]. In particular, employers continue to express doubts that university education does indeed lead to professional

competence, claiming that it fails to provide students with the skills they actually require in the workforce [17]. At the same time, undergraduate employment rates are slumping [7], which often leads to further delays in the time it takes students to find work upon graduation, in turn leading to requirements for further professional training [10].

Until recently, much of this debate has been poorly supported by evidence and data. Claim and counterclaim prevail, but a large amount of the data supplied has been *ad hoc*, or cherry picked to support vested interests [20, 9]. However, with the rise of online job advertisements it has become possible to collect data about what potential employers demand in the workplace. A number of datasets can now be created, using data collected from web platforms such as LinkedIn¹, SEEK² and Monster³. Indeed, vendors such as Burning Glass (BG) technologies⁴ now market aggregation services and data that can be used to understand changing trends in the workforce. The next sections provide a brief overview of the ways in which this data can be used.

1.1 Curriculum analytics

Many attempts have been made to understand what gaps there might be in the curriculum offerings of educational institutions. For example, Knight and Yorke [13] describe the Skill plus project as an attempt to manually audit the university curricula for four universities and 17 departments. Trying to find curriculum gaps, Davis et al. [5] conducted a survey of graduates, Lang et al. [15] surveyed industry representatives, Lempp and Seale [16] conducted a study among health students. However, the manual and resultingly not sufficient scalability of this work, has limited the use of this work in linking to workforce needs [11].

¹<https://www.linkedin.com/jobs/>

²<https://www.seek.com.au/>

³<https://www.monster.com/>

⁴<https://www.burning-glass.com>

Aleksandr Gromov, Andrei Maslennikov, Nikolas Dawson, Katarzyna Musial and Kirsty Kitto "Curriculum profile: modelling the gaps between curriculum and the job market" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 610 - 614

More recently we start to see data science techniques used to overcome these limitations. For example, attempts have been made to infer curriculum information from student performance and interactions with the curriculum [20, 21], but not the skills offered in the curriculum itself. *Skills based curriculum analytics* [12] makes use of Natural Language Processing (NLP) to map curriculum documents to a defined skills taxonomy, in this case for the purpose of recognising prior learning. A number of other works have made use of NLP in analysing curriculum [1, 8, 24, 19, 23], but only for specific subsets of curriculum (normally computer science). Thus, opportunities are emerging for automating the analysis of curriculum, but how might this be linked with labour market demand?

1.2 Connecting market demand and curriculum

Educational institutions have recognised their role in supplying market demand by offering various courses that prepare graduates to enter the workforce [4]. This transforms the problem of mapping curriculum into one of alignment with labour market requirements [14]. One of the most common methods applied to this task adopts industry skill frameworks and then formulate graduate attributes which match market expectations [14, 22]. This method allows curriculum managers and developers to see gaps, and work to align existing and new courses towards market expectations. It also helps students to plan their study course according to their career desires [25]. However, both frameworks and curriculum are living documents that adapt to the environment, and there is a time lag between acknowledging and implementing new skills and technologies into frameworks and curriculum. Furthermore, this curriculum mapping task is usually completed manually (see section 1.1, which makes it time-consuming, tedious, and prone to mistakes. Worse still, some industries may rarely update their industry frameworks, while other industries might not even have a formal listing of their skill requirements [6].

Overall, while many institutions make use of industry advisory bodies, market reports, etc. to map their curriculum by hand, we are yet to find examples of curriculum mappings to workforce requirements that are supported by the emerging large scale employment datasets that are becoming common in the field. This is the gap that we seek to address here.

1.3 Research questions and contribution

This work aims to use automated methods to map gaps between the subset of the curriculum taught at one institution and labour market demands. We will do this by asking the following three research questions:

- RQ1:** What is the skills profile for a complete institution?
- RQ2:** How can we explore the gap between university curriculum and market demand?
- RQ3:** How can we differentiate the match between contrasting curriculum pathways and labour market demand?

Our contributions include: (i) a preliminary method for automatically constructing a curriculum profile for an institution (ii) a way to compare subsets of curriculum within that institution (iii) a method for finding gaps between closely

related set of degree programs and the job market.

2. CURRICULUM PROFILE

This paper introduces the concept of a university curriculum profile. We make use of the BG ontology, which provides a static set of skills that can be consistently mapped into a range of different higher level clusters to extract information about what mix of skills is being taught across an entire institution. We chose the University of Technology Sydney (UTS) curriculum as a data source, which consists of 486 degree programs (termed courses) offered across 9 faculties at the undergraduate and postgraduate levels. In total, there are 3,739 subjects offered across these degrees. Information about the curriculum can be obtained using the UTS handbook (<https://www.handbook.uts.edu.au/>).

2.1 Method

This section discusses three experiments that have been performed, each designed to extract information about the skills taught at UTS. We start with a course profile across the entire curriculum of UTS in a bid to respond to RQ1 (Section 2.1.1), before performing a deeper analysis of the data analysis curriculum offerings at the same institution to respond to RQ2 (Section 2.1.2). Finally, in Section 2.1.3 we determine how well aligned these data analysis offerings are with local labour market demand, so responding to RQ3.

2.1.1 The UTS curriculum profile

For the first part of our analysis, we performed a skills analysis of the entire UTS curriculum, with a view towards developing a skills profile across the university. This then enables us to drill into sub-profiles for specific degree programs, demonstrating their similarities and differences. Our analysis implemented the following steps:

- STEP 1:** We scraped the UTS curriculum handbook.
- STEP 2:** Subject names and descriptions were mapped to lists of skills using the BG content tagger.
- STEP 3:** Skills were mapped to higher level skill clusters and families according to the BG Skill Ontology.

Drilling into the skill cluster families tagged by BG makes it possible for us to start exploring the distribution of skills taught by each Faculty at UTS. Encouragingly, this method reveals that the majority of the skills developed by each Faculty are in sensible skill cluster family domains, with some spread into other families, in explainable patterns. Thus, the Faculty of Engineering and Information Technology (FEIT) teaches 100% of all *Engineering* and 97% of all *Information Technology* skills covered at UTS; the Faculty of Health (Health) teaches the largest proportion of *Health Care* skills, followed by the Graduate School of Health (GSH); and the Faculty of Business covers the *Business* and *Finance* skills.

2.1.2 Within the data analysis curriculum

For our second analysis, we decided to perform a deep dive into a subset of the UTS curriculum. We chose to explore the Data Analytics related degrees available across UTS. This decision was based upon the existence of three potentially competing degrees that are currently offered at UTS

MDSI: Master of Data Science and Innovation (MDSI)

MIT: Master of Information Technology (Data Analytics)
MBA: Master of Business Administration (Data Analytics)

We sought to explore how much overlap existed in the curriculum associated with these three degrees, and whether there was a possibility that inefficiencies could be identified where the same skills were being taught across multiple subject offerings. We implemented this analysis by following a similar sequence to that presented in Section 2.1.1. Each of the three selected courses consists of a set of compulsory core subjects and an optional choices. We performed two separate analyses, extracting a skills profile for two different course structures: one using just the core subjects required for each of the three courses (core), and a second analysis that added the Data Analytics subject selections for that degree (data). We chose the IT, Business and Analytics skill cluster families for a further investigation of possible reasons of skills changes, because these are the claimed focus of the chosen degrees.

After skill profiling a number of patterns can be noted, for example, the MIT has the most complete coverage in the *Information Technology* skill cluster family, but has almost no coverage of the *Analytics* skills cluster family. However, adding the Data Analytics subject choices to the Core leads to an increase of skills in the *Analytics* skill cluster family (almost to the point where it has the same number of skills as the entire data science-oriented MDSI degree). Similarly, the Core MBA subjects cover all three skill families, but selecting more specific Data subjects leads to a growth in the number of skills in *IT* and *Analytics* skill cluster families. Finally, as expected given the exclusion of the full set of optional subjects no change is observed for the MDSI when expanding with Data subjects.

2.1.3 Finding a gap between market demand and curriculum offerings

For our final study, we combined UTS curriculum data with data about skills sought in the Australian job market:

STEP 1: We selected the top 10 Data Science and Analytics (DSA) skills, found by Dawson et al. [6] and required by the market, and checked if they exist in UTS curriculum and MIT, MDSI and MBA courses.

STEP 2: Then, we selected the top 10 DSA skills that showed the highest growth in the market and cross-checked to see if the UTS curriculum adapts to these rapid market changes.

STEP 3: After that, we selected three DSA Occupations from the BG ontology, retrieved the skills linked with these occupations in the BG ontology and compared them with the skills covered by the UTS curriculum.

Encouragingly, all of the top-10 DSA skills exist in the curriculum. However, some skills are missing from the selected Data Analytics courses. Overall, selected courses cover most of the demanded DSA skills.

At the same time, only four skills with the highest compound annual growth rate (CARG) in 2019 [6] exist in the resulting skills profile. Other technologies and tools, such as Blockchain, TensorFlow, Internet of Things, are missing in the curriculum we analysed. However, some of these skills

are yet to be integrated into the skills clusters and families of the BG ontology, which points to their very recent emergence. This gap points to an opportunity for UTS to identify rapidly growing skills that it considers beneficial to deliver: a curriculum gap that could be rectified.

Finally, we retrieved three DSA occupations (Data Analyst, Data Scientist and Business Intelligence Analyst) and their skills from the BG occupation ontology comparing them with the skills profiles for the MIT, MDSI and MBA (see Table 1 for the core subject selections). The results mirror those obtained in the previous section.

Overall, none of the UTS courses has more than 50% of the skills taught that required in our three selected occupations which potentially demonstrates a gap between university curriculum and market demand.

Occupation	Course	Both exist	Only in Occupation	Only in Course
Data Scientist	MDSI	27	73	97
	MIT	42	58	287
	MBA	37	63	153
Data Analyst	MDSI	20	80	104
	MIT	42	58	287
	MBA	34	66	156
Business Intelligence Analyst	MDSI	17	83	107
	MIT	35	65	294
	MBA	32	68	158

Table 1: Three DSA occupations from BG ontology with the number of skills existing and not existing in three selected UTS courses.

3. TOWARDS A CURRICULUM PROFILE

A number of findings about the curriculum taught at UTS emerge from the preliminary curriculum profile presented in the previous section. Firstly, there is a gap between BG skill ontology and the UTS curriculum profile. However, UTS develops knowledge not just software skills. The analysis shows only seven families are more than 50% covered by the UTS curriculum (Business, Economics, Engineering, Environment, Legal, Media and Science). At the same time, the most well-presented in the curriculum IT and Health families cover only 25% and 38% respectively. The majority of skills in these families are missing. However, the reasons for the gap are interesting in themselves, including:

- 1. Novelty skills:** there is a lag between new skills emerging (e.g. TensorFlow, WebAssembly) in the market and their incorporation into curriculum offerings.
- 2. Legacy skills:** in contrast, some skills (e.g. COBOL, ALGOL and the early versions of Microsoft Server) are present in the ontology but not taught at UTS.
- 3. Generalised knowledge:** the role of universities is larger than that of simple skill development. We see evidence that UTS is developing generalisable knowledge, rather than the more specific skills.

Another finding from our approach is that the faculties at UTS do appear to have specialisations which largely match the subject materials we would expect to see taught in the faculty. Thus, for example, the Business Faculty has a focus on “Finance” and “Business” (87% and 73% of all UTS skills in this families), Faculty of Engineering and IT includes “IT”

and “Engineering” (97% and 100% of all UTS skills in this families) and Science Faculty prepares students in “Science and Research” and “Environment” (100% and 77% of all UTS skills in this families). Overall, the UTS curriculum profile demonstrates that the university tends to develop knowledge, not just skills and an ability to use specific tools.

3.1 Data Analysis projection on different courses

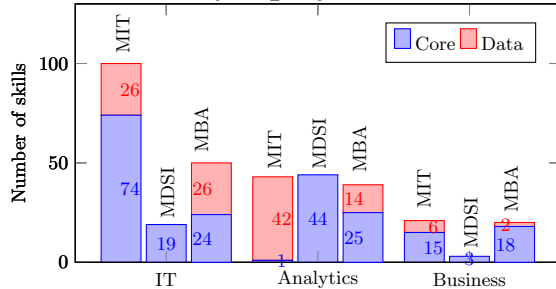


Figure 1: Number of skills in three courses (Core and Data selections) for three BG skill families.

All three selected DSA courses have unique skill profile and prepare different types of Data Experts. By selecting three Data related courses from three different faculties, we wanted to find out if the courses, offered across UTS, are repetitive and faculties lack collaboration. The results in Figure 1 show that each course has a different flavour which aims to develop different fundamental skills. The MDSI focuses on Analytical skills, the MIT focuses upon IT skills and the MBA is giving fundamental knowledge in IT, Business and Analytics. However, expanding the Core subject selection to DSA major, we see that: (i) MBA almost stopped developing business skills and focused on Analytics, which also resulted in the number of IT skills increasing. (ii) MIT becomes far more aligned with the Data Analytics skill cluster, with an accompanying increase in IT and Business skill development (iii) MDSI did not change the curriculum profile at all, which is related more to the extremely open course structure of the degree and the fact that this could not be captured by our method. Overall, each course prepared almost the equal number of skills in Analytics (39 for MBA, 43 for MIT, 44 for MDSI). However, the content and the student pathway are not the same. Students from MDSI are focusing on Data Analytics, as expected from a Data Science course, but MIT students have a strong background in IT, supported by Analytics. Finally, MBA students are well-rounded specialists in IT, Business and Analytics.

3.2 Finding a curriculum gap

The first gap revealed in section 2.1.3 illustrates a disconnection between the understanding of graduate capabilities possessed by universities and the market. While some skill sets are too narrow to be useful to a graduate (e.g. Atlas.ti and Alteryx), others are popular on the job skills for data analysts and data scientists (e.g. SQL and Git). A domain expert is required to make the distinction between these skills, but our method of building curriculum profiles could help with decision support, showing up skills that are essential but still largely unrepresented in the curriculum.

The second gap we found is a lag between the appearance of

a new area of knowledge in the labour market and its introduction to the curriculum. This gap should be distinguished from the emergence of new tools alone, although there can be some overlap (e.g. with TensorFlow and Apache Spark). More general skills like Deep Learning, Data Lakes and Random Forests also feature in this list. Similarly, there is a need for Internet of Things and Blockchain specialists which the UTS curriculum is yet to respond to. This second gap is potentially more dangerous because it shows incapability to cover new areas of knowledge in time. However, more analysis is required to investigate the actual demand for these emerging skills. For example, while the CAGR can be very high, this can be achieved by doubling the demand for a skill that was previously only advertised for twice in a time period. Care must be taken to disentangle growth from absolute demand, a task that we reserve for future work.

4. CONCLUSIONS AND FUTURE WORK

In this article, we introduced the university curriculum profile that allowed us to explore the skills taught across an entire university, and to establish that the faculties at UTS do indeed teach the skills we expect. It also enabled us to demonstrate the existence of a potential gap in what was taught, but which was explained by unearthing the too specific and technology dependent nature of many skills in the BG ontology. This gap was explained with the observation that universities should be developing graduates who can generalise knowledge from their skillsets, not just make use of a highly specific tool, and was therefore identified as not critical.

This work can be extended in several ways. Firstly, it will be important to find ways of representing the complexity of a curriculum structure using more than counts. Many of the potential gaps our analysis identified turned out to be understandable once we looked deeper into the skills that were not being taught (Section 2.1.3). Secondly, the analysis can and should be extended beyond the DSA degrees considered here to explore what differences may result from using different skill sets. Third, finding the curriculum gaps will benefit from the development of automated tools for finding changes in labour market demand. Such instruments could be tuned to track changes in real-time and provide historical data for more in-depth analysis of the university curriculum and its development. Finally, the current method of extracting skills from the curriculum cannot identify differences between novice and advanced skills. It is essential that we develop additional tools for evaluating these different levels of skill proficiency.

We believe that the method of profiling curriculum developed here will help institutions to adjust existing courses or initialise new ones as required by the market. It will also help students to choose more effective learning paths according to the market demand. As such, it has the potential to help institutions improve outcomes for all learners.

5. ACKNOWLEDGMENTS

We acknowledge the support of Burning Glass in provisioning the API tools that were used in this study.

References

- [1] ACM/IEEE-CS Joint Task Force on Computing Curricula. Computer science curricula 2013. Technical report, ACM Press and IEEE Computer Society Press, December 2013. URL <http://dx.doi.org/10.1145/2534860>.
- [2] T. Brock. Young adults and higher education: Barriers and breakthroughs to success. *The future of children*, pages 109–132, 2010.
- [3] N. Chhinzer and A. M. Russo. An exploration of employer perceptions of graduate student employability. *Education+ Training*, 2018.
- [4] M. Clarke. Rethinking graduate employability: The role of capital, individual attributes and context. *Studies in Higher Education*, 43(11):1923–1937, 2018.
- [5] R. Davis, S. Misra, and S. Van Auken. A gap analysis approach to marketing curriculum assessment: A study of skills and knowledge. *Journal of Marketing Education*, 24(3):218–224, 2002.
- [6] N. Dawson, M.-A. Rizoio, B. Johnston, and M.-A. Williams. Adaptively selecting occupations to detect skill shortages from online job ads. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1637–1643, Dec 2019.
- [7] B. De La Harpe, A. Radloff, and J. Wyber. Quality and generic (professional) skills. *Quality in Higher Education*, 6(3):231–243, 2000.
- [8] E. Durant, J. Impagliazzo, S. Conry, R. Reese, H. Lam, V. Nelson, J. Hughes, W. Liu, J. Lu, and A. McGettrick. Ce2016: Updated computer engineering curriculum guidelines. In *2015 IEEE Frontiers in Education Conference (FIE)*, pages 1–2. IEEE, 2015.
- [9] B. G. T. (Firm)(US). Moving the goalposts: How demand for a bachelor’s degree is reshaping the workforce. 2014. URL https://www.burning-glass.com/wp-content/uploads/Moving_the_Goalposts.pdf.
- [10] J. Keating et al. *Current vocational education and training strategies and responsiveness to emerging skills shortages and surpluses*. National Centre for Vocational Education Research, 2008.
- [11] J. Keevy and B. Chakroun. Level-setting and recognition of learning outcomes: The use of level descriptors in the twenty-first century. Technical report, 2015. URL <http://unesdoc.unesco.org/images/0024/002428/242887e.pdf>.
- [12] K. Kitto, N. Sarathy, A. Gromov, M. Liu, K. Musial, and S. Buckingham Shum. Towards skills-based curriculum analytics: Can we automate the recognition of prior learning? In *Proceedings of the 10th International Conference on Learning Analytics and Knowledge*. ACM, 2020.
- [13] P. T. Knight and M. Yorke. Employability through the curriculum. *Tertiary education and management*, 8(4): 261–276, 2002.
- [14] B. R. Konsky, A. Jones, and C. Miller. Embedding professional skills in the ict curriculum. In *ASCILITE-Australian Society for Computers in Learning in Tertiary Education Annual Conference*, pages 883–887. Australasian Society for Computers in Learning in Tertiary Education, 2013.
- [15] J. D. Lang, S. Cruse, F. D. McVey, and J. McMaster. Industry expectations of new engineers: A survey to assist curriculum designers. *Journal of Engineering Education*, 88(1):43–51, 1999.
- [16] H. Lempp and C. Seale. The hidden curriculum in undergraduate medical education: qualitative study of medical students’ perceptions of teaching. *Bmj*, 329(7469):770–773, 2004.
- [17] K. Lowden, S. Hall, D. Elliot, and J. Lewin. Employers’ perceptions of the employability skills of new graduates. *London: Edge Foundation*, 2011.
- [18] G. Mason, G. Williams, and S. Cranmer. Employability skills initiatives in higher education: what effects do they have on graduate labour market outcomes? *Education Economics*, 17(1):1–30, 2009.
- [19] Y. Matsuda, T. Sekiya, and K. Yamaguchi. Curriculum analysis of computer science departments by simplified, supervised lda. *Journal of Information Processing*, 26: 497–508, 2018.
- [20] X. Ochoa. Simple metrics for curricular analytics. In *Proceedings of the 1st Learning Analytics for Curriculum and Program Quality Improvement Workshop*, 2016.
- [21] Z. A. Pardos, Z. Fan, and W. Jiang. Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. *User Modeling and User-Adapted Interaction*, 29(2):487–525, 2019.
- [22] R. S. Pillutla and M. Narayana. Framework integrating multiple dimensions of competency and related pedagogies: A case for it industry. In *2013 12th International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 1–8. IEEE, 2013.
- [23] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.
- [24] T. Sekiya, Y. Matsuda, and K. Yamaguchi. Curriculum analysis of cs departments based on cs2013 by simplified, supervised lda. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge*, pages 330–339. ACM, 2015.
- [25] L. S. Xun, S. Gottipati, and V. Shankararaman. Text-mining approach for verifying alignment of information systems curriculum with industry skills. In *Information Technology Based Higher Education and Training (ITHET), 2015 International Conference on*, pages 1–6. IEEE, 2015.

Assessing Student Contributions in Wiki-based Collaborative Writing System

Tianyu Hu
University of Science and
Technology of China
hty98@mail.ustc.edu.cn

Guangzhong Sun
University of Science and
Technology of China
gzsun@ustc.edu.cn

Zhongtian Xu
University of Science and
Technology of China
xuzt@mail.ustc.edu.cn

ABSTRACT

In recent years, Wiki has been proved effective for collaborative learning in modern education. As a typical collaborative writing system, Wiki empowers students in generating, modifying and structuring their own contents. Some courses may include these collaborative assignments like writing a wiki page as part of assessment. But for teachers, it is difficult to assess the quality of student contributions, because the final result of project is made up of edits from different students. In this paper, we propose a content-based model, OSEAN(Order-Sensitive Edit Assessing Network) to better address this problem. OSEAN can represent and predict students edits' quality by extracting semantic features from edit pairs. Experiment results show that OSEAN has the highest AUPRC on Wikipedia edit quality classification task in all tested methods. Furthermore, OSEAN can handle reversed edit pairs correctly, which often happens when one student undoes previous student's edit.

Keywords

Natural Language Processing,Assessment,Collaborative Learning,Sequence Modeling,Wikipedia,Crowdsourcing

1. INTRODUCTION

In recent years, the use of modern information and communication technologies in education has been widely studied[10]. Thanks to the rapid development of web technology, higher level of collaborative learning becomes easier. Among these web applications, wiki attracted attention for enabling students work together. According to the definition on Wikipedia, wiki is a knowledge base website on which users collaboratively modify and structure content directly from a web browser. These inherent characteristics of wiki technology encourage students collaborate to create their own contents[3].

However, assessing student contributions in a wiki project can be difficult. This is because that students not only add

Tianyu Hu, Guangzhong Sun and Zhongtian Xu "Assessing Student Contributions in Wiki-based Collaborative Writing System" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 615 - 619

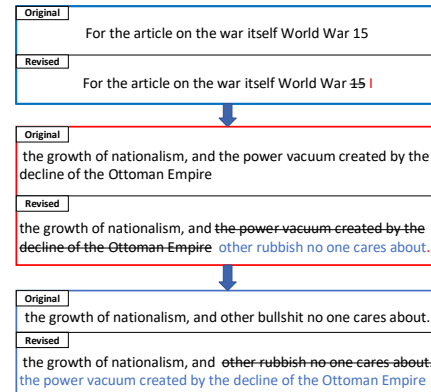


Figure 1: Example of revision history from Wiki page: Causes of World War I. We select 3 continuous versions and compare the differences. Edit 1 fixed an error in the page. Edit 2 deleted some words and added some offensive words. Edit 3 did a revert operation to eliminate vandalism information introduced by revision 2.

contents to the project, but also revise or delete contents which are added by others. Since reprocessability plays a key role in evaluation of student works[6], we should assess student contributions from the entire process of wiki project. If teachers only evaluate everyone's contribution from the final state of the project, then some important behavior information can be lost. Figure 1 gives an example of page revision history. In this work, we care about the quality of students contribution in the project, so we need to evaluate the quality of each edit. A wiki project usually consists of many edits, which brings a lot of works to teachers. Therefore, we want to evaluate the quality of edits in an automated way. To predict new edits' quality, two types of methods are proposed. Content-based methods extract features from the content of edits. ORES[5] and Stiki[11], web services provided by Wikimedia team, use linguistic features to compute the probability that a specific edit is damaging. StRE[8] utilizes deep neural network and achieves a high accuracy. On the other hand, content-independent methods, e.g. Interank[12], treat edit as the interaction between user and project(page).

While each edit is a pair of sequences before and after an edit, a new question arises: Does the order of pair matter? The order of edit pairs represents the direction of contents

evolution. If a model can truly predict the quality of an edit, it should generate a opposite label if we reverse the edit pair.

To better predict new edits' quality and handle the order of edit pair. In this work, we propose OSEAN(Order-Sensitive Edit Assessing Network), a content-based edit quality prediction model. OSEAN extracts each dissimilar part of two sentences and learn the vector representations for two parts. To handle the order of edit pair, we utilize the subtract result between two parts as the final representation of the entire edit.

2. METHODOLOGY

2.1 Problem Formulation

An edit $P = \{S, T\}$ on a particular page is a pair of original sentence S and revised sentence T . Each sequence is represented as a fixed length character sequence.

$$\begin{aligned} S &= \{S_1, S_2, \dots, S_M\} \\ T &= \{T_1, T_2, \dots, T_M\} \end{aligned}$$

where M is the length that can be manually set. Our task is to find a page-specified scoring function that maps each edit to a binary label:

$$f_{page} : P \rightarrow L, L \in \{0, 1\} \quad (1)$$

2.2 Model Architecture

Figure 2 gives an overview of OSEAN. We will introduce each steps in the model below.

Character Embedding. The first layer performs a character-level look-up where each character is represented as a d -dimension vector. The edit pair is converted to two matrices of dimension $m \times d$.

Convolution Step. After the character-level embedding, the sequences of embedded characters is provided as inputs of convolution layer, which computes an 1-D convolution over the embedded sequences. A convolution operation involves a filter with size h :

$$c_i = \tanh(w_c \cdot x_{i:i+h-1} + b_c) \quad (2)$$

As a result, each sentence is represented as a feature map of dimension $l \times d$, where $l = m - h + 1$.

Dissimilar Part Extraction. Since an edit is changes of page contents, the dissimilar part of two sequences should have higher weights on qualities. We utilize the method from [9]. In our model, the semantic unit of the sequence is the combinations of characters after the convolution operation, and we only care about the dissimilar part. To determine which part is *dissimilar*, we need to check whether a unit is semantically covered by another sequence.

First, we compute the similarity matrix $A_{L \times L}$ for feature maps C_S and C_T after the convolution step, each element

$a_{i,j} \in A$ is the cosine similarity between unit $C_{S,i}$ and $C_{T,j}$.

$$a_{i,j} = \frac{C_{S,i}^\top C_{T,j}}{\|C_{S,i}\| \|C_{T,j}\|} \quad (3)$$

Then we use the similarity matrix to calculate the semantic cover of $C_{S,i}$ by combining all units in the other sequence C_T .

$$\text{cover}(C_{S,i}, C_T) = \frac{\sum_{j=0}^L a_{i,j} C_{T,j}}{\sum_{j=0}^L a_{i,j}} \quad (4)$$

The result $\hat{C}_{S,i} = \text{cover}(C_{S,i}, C_T)$ can be used to calculate the proportion α of unit $C_{S,i}$ that is present in the other sequence. The value of α is the cosine similarity α of $C_{S,i}$ and $\hat{C}_{S,i}$. So the dissimilar part's can be defined as $1 - \alpha$. The dissimilar part $D_{S,i}$ for feature map unit $C_{S,i}$ is:

$$\alpha_i = \frac{C_{S,i}^\top \hat{C}_{S,i}}{\|C_{S,i}\| \|\hat{C}_{S,i}\|} \quad (5)$$

$$D_{S,i} = (1 - \alpha_i) C_{S,i} \quad (6)$$

After performing the above calculations for all units in C_S and C_T , we get two dissimilar parts D_S and D_T .

Edit Representation. We use a weight-sharing fully connected layer(FCL) to generate representation vectors E_S, E_T for each sequence. To obtain the final representation E_{final} for the whole edit, we perform a **subtract** operation on E_S and E_T . The edit vector E_{final} is used for quality classification with a sigmoid activation.

$$E_S = W_0 D_S + b_0, E_T = W_0 D_T + b_0 \quad (7)$$

$$E_{\text{final}} = E_S - E_T \quad (8)$$

$$r = \text{sigmoid}(W_1 E_{\text{final}} + b_1) \quad (9)$$

Here, r is considered to be the possibility that the edit P to be a beneficial edit.

2.3 Order of Edit Pair

Consider an edit $P = (S, T)$, we assume P to be a beneficial edit and labeled as 1. If we reverse the order of the edit pair, the label of the reversed edit $P' = (T, S)$ should also be flipped, meaning P' has a label 0. This is because the reverted operation on a beneficial edit should be considered to be a damaging edit. If the order of the pair can not be handled correctly, the model is very likely to classify two opposite edit P and P' to be the same label.

We give the definition of *order-sensitive* here:

Definition 1 (order-sensitive). *A model is order-sensitive if for most edit pairs, it satisfies: the model gives two opposite labels for edit P and its reversed version P' .*

Obviously, an ideal edit quality prediction model should be order-sensitive. Our proposed model is *perfectly* order-sensitive under ideal conditions which can be proven mathematically and also performed well in the experiment:

3. EXPERIMENTS

In this section, we conduct experiments to answer following questions:

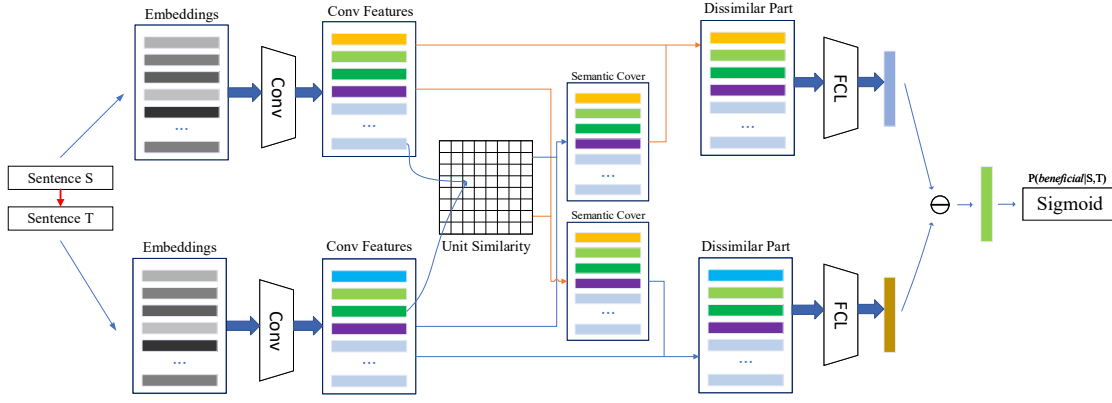


Figure 2: An overall architecture of OSEAN(Order-Sensitive Edit Assessing Network)

RQ1 Can our proposed model outperform state-of-the-art edit quality prediction methods?

RQ2 If the order of edit pairs is reversed, how the performance of all experimented methods will change? Can our model still maintain the best performance?

RQ3 If we add the order information to model training process by performing data augmentation, will model performance get improved?

3.1 Experiments Setup

3.1.1 Dataset

We evaluate model’s performance on the data extracted from Wikipedia page revision histories. As Wikipedia is the most widely used collaborative writing system in the world, experiments on this system can verify the effectiveness of our model. Page histories are divided into three categories:

1. CS: Pages containing top 147 pages with the highest number of edits related to computer science in English Wikipedia as of June 2017[8].
2. EN/ZH: Pages containing top 68/55 pages in the whole English/Chinese Wikipedia as of June 2019.

The number of samples in each category is reported in 1.

3.1.2 Computation of Edit Quality and Label

The basic idea is that if changes introduced by an edit is preserved in several subsequent edit, then the edit is considered to be beneficial. Otherwise, if the changes is reverted, then the edit is damaging. [1] and [2] give a formula to compute the proportion of preserved changes. We follow the approach and use a average value to compute the edit quality.

Consider a particular page and denote its k -th revision (i.e., the state of the article after the k -th edit) as v_k . Let $d(u, v)$ be the Levenshtein distance[7] between two sentences. We define the quality of edit k from the perspective of the article’s state after $\ell \geq 1$ subsequent edits as:

$$q_{k|\ell} = \frac{d(v_{k-1}, v_{k+\ell}) - d(v_k, v_{k+\ell})}{d(v_{k-1}, v_k)} \quad (10)$$

Samples	CS	EN	ZH	Total
# Total	2377732	285365	122748	2785845
# $q \geq 0$	1402596	190924	88621	1682141
# $q < 0$	975136	94441	34127	1103704

Table 1: Number of samples for each category in dataset

We compute the average value over several future revisions:

$$q_k = \frac{1}{L} \sum_{\ell=1}^L q_{k|\ell} \quad (11)$$

We set $L = 10$ to compute the final edit quality in data pre-processing. Each edit’s quality is automatically computed and labeled as *damaging* if the quality score $q < 0$, and labeled as *beneficial* if $q \geq 0$.

3.1.3 Competing Approaches

We compare OSEAN with some existing methods:

Average The average approach always outputs the ratio of good edit on the training set as the predict probability.

ORES The Objective Revision Evaluation Service (ORES)[4, 5] is an open-source classifier system developed by researchers at the Wikimedia Foundation.

Interank Interank[12] uses matrix factorization method to learn editor’s ability and page’s difficulty based on the page’s edit history.

StRE StRE(Self Attentive Revision Encoder)[8] is a deep learning based method which combines word level signals as well as character level signals.

ABCNN Attention Based Convolutional Neural Network (ABCNN)[13] integrates attention into CNNs for general sentence pair modeling tasks. We use ABCNN-2 for our edit classification task.

3.1.4 Evaluation

To compare the performance of models, we set up a classification task to predict if an edit is beneficial or not. For

Model	CS	EN	ZH	Total
Average	0.733	0.714	0.814	0.745
Interank	0.448	0.427	0.352	0.436
ORES	0.832	0.852	0.838	0.834
StRE	0.898	0.877	0.884	0.890
ABCNN	0.899	0.912	0.938	0.905
OSEAN	0.946	0.945	0.952	0.947

Table 2: Results on Wikipedia dataset

Model	Ori-Test	On Test Rev-Test	Diff	Ori-Train	On Train Rev-Train	Diff
Average	0.745	0.305	-0.440	0.745	0.270	-0.475
ORES	0.835	0.316	-0.519	0.931	0.286	-0.645
StRE	0.890	0.404	-0.486	0.923	0.345	-0.578
ABCNN	0.905	0.497	-0.408	0.924	0.450	-0.474
OSEAN	0.948	0.755	-0.193	0.993	0.957	-0.036

Table 3: Results for reversed pair experiment. Ori-* denotes the original set, Rev-* denotes the reversed set.

each example, we compute the quality score based on the revision history and assign each example a binary label.

For each particular page, we split the edits on the page randomly into train/validation/test set with ratio 80%/10%/10% and train models. Page-specific models are evaluated and we use the average AUPRC in each category as the final metric which is consistent to previous works[12, 8].

3.2 Basic Experiment (RQ1)

We evaluate OSEAN on the original test set to answer the first question. Table 2 presents the average AUPRC value for each category in original test set. OSEAN has the highest AUPRC and is 4.6% higher than the next-best method, proving the effectiveness of our proposed model.

3.3 Reversed Pair Experiment (RQ2)

In this experiment, we use the same train and validation set as before. For test set, we design two settings:

1. On Test : Trained models are evaluated on original and reversed *test* set.
2. On Train: Trained models are evaluated on original and reversed *train* set.

A reversed dataset is generated by reversing every edit pair and flipping the labels in the original set. According to the definition, an order-sensitive model should have similar performance on original and reversed set. Thus, the difference in AUPRC can be used as a criterion to determine whether the model is order-sensitive. We use average AUPRC of all pages as metric.

Results. Experiment results are reported in Table 3. Interank model is not tested because the reversed sample is anonymous which can not be processed by Interank. Performance of all models drops when classifying reversed pairs. OSEAN has the smallest decline which is 52.7% lower on test and 92.4% lower on train than the next-best method. OSEAN has the smallest performance decline and highest AUPRC on reversed set in both settings, proving that our model can handle reversed edit pairs correctly.

3.4 Training with Augmentation (RQ3)

In this experiment, we use training set with data augmentation to train models. For each example $P = (S, T)$ with label ℓ in training set, we add a reversed example $P' = (T, S)$ with

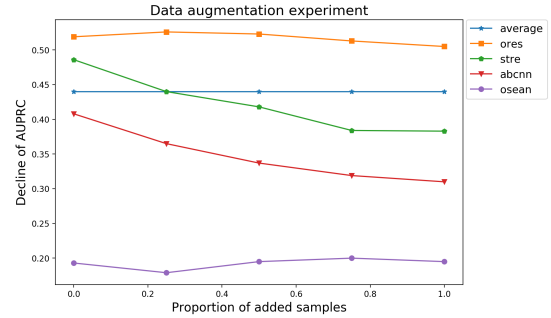


Figure 3: Results on decline of AUPRC with different proportion of data augmented.

the opposite label ℓ' to the training set. Models are trained on augmented training set and evaluated on both original and reversed test set. We train models with five different cases (i.e. when 0%/25%/50%/75%/100% of reversed training pairs are added). We want to know if data augmentation allows models to learn the information of pair order and empowers models to be order-sensitive.

Results. Performance decline with different rates of data augmentation is reported in Figure 3. As more data is added, the performance gap between the original and reversed test set is also declined. The narrowing of the gap proves that data augmentation can indeed make models more order-sensitive. However, even with 100% data augmentation, the performance gap for all baseline methods is still large, and gap for OSEAN is 37.4% lower than the next-best method.

4. CONCLUSION

In this paper, we present OSEAN, a content-based model for assessing edit quality in wiki-based writing system. Our method utilizes the convolution network to find semantic differences between previous and revised sentences, which can represent an edit. Experimental results on page revision histories from Wikipedia demonstrate that our model can effectively predict new edits' quality. Therefore, we can more accurately determine the quality of student contributions in the project.

5. ACKNOWLEDGMENT

This work is supported by Youth Innovation Promotion Association of CAS. Guangzhong Sun is the corresponding author of this work.

6. REFERENCES

- [1] B. T. Adler and L. De Alfaro. A content-driven reputation system for the wikipedia. In *Proceedings of the 16th international conference on World Wide Web*, pages 261–270. ACM, 2007.
- [2] B. T. Adler, L. De Alfaro, I. Pye, and V. Raman. Measuring author contributions to the wikipedia. In *Proceedings of the 4th International Symposium on Wikis*, page 15. ACM, 2008.
- [3] M. Cole. Using wiki technology to support student engagement: Lessons from the trenches. *Computers & education*, 52(1):141–146, 2009.
- [4] A. Halfaker and R. S. Geiger. Ores: Lowering barriers with participatory machine learning in wikipedia. *arXiv preprint arXiv:1909.05189*, 2019.
- [5] A. Halfaker and D. Taraborelli. Artificial intelligence service “ores” gives wikipedians x-ray specs to see through bad edits, 2015.
- [6] W. He and L. Yang. Using wikis in team collaboration: A media capability perspective. *Information & Management*, 53(7):846–856, 2016.
- [7] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.
- [8] S. Sarkar, B. P. Reddy, S. Sikdar, and A. Mukherjee. Stre: Self attentive edit quality prediction in wikipedia. *arXiv preprint arXiv:1906.04678*, 2019.
- [9] Z. Wang, H. Mi, and A. Ittycheriah. Sentence similarity learning by lexical decomposition and composition. *arXiv preprint arXiv:1602.07019*, 2016.
- [10] M. Warschauer. Computer-mediated collaborative learning: Theory and practice. *The modern language journal*, 81(4):470–481, 1997.
- [11] A. G. West, S. Kannan, and I. Lee. Stiki: an anti-vandalism tool for wikipedia using spatio-temporal analysis of revision metadata. In *Proceedings of the 6th International Symposium on Wikis and Open Collaboration*, page 32. ACM, 2010.
- [12] A. B. Yardim, V. Kristof, L. Maystre, and M. Grossglauser. Can who-edits-what predict edit survival? In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2604–2613. ACM, 2018.
- [13] W. Yin, H. Schütze, B. Xiang, and B. Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272, 2016.

EAnalyst: Toward Understanding Large-scale Educational Data

Tao Huang¹, Zhi Li², Hao Zhang¹, Huali Yang^{1,*}, Hekun Xie²

National Engineering Research Center for E-learning, Central China Normal University, Wuhan, China

¹{tmht, zhanghao, yanghuali}@mail.ccnu.edu.cn

²{zhili, xiehekun}@mails.ccnu.edu.cn

ABSTRACT

We present an educational data collecting, mining and analyzing system, EAnalyst, for learners in the K12 period, providing highly intellectual personalized analysis and recommendations for learners. EAnalyst consists of preprocess module, analysis module, dashboard module and recommendation module. To assess target learner's knowledge proficiency better, we extend the current deep knowledge tracing model to achieves the goal of performance predicting. The results on both open dataset and our platform dataset demonstrate the effectiveness of our model run on our platform.

Keywords

E-Learning; Personalized Analysis; Data Mining

1. INTRODUCTION

The rapid development of information technology has helped the "learner-centered" teaching mode attracting more and more attention. With the assistance of big data analysis and artificial intelligence, promoting large-scale data-driven personalized learning analysis has become realistic. EAnalyst is a system whose main goal is to provide intelligent, personalized, and novel assistance to learners.

To meet the increasing needs of personalized learning [1], some existing work focuses on single work or test of a target learner [2] without continuous tracking and analysis of the whole learning process. Chronological data contain hidden patterns that are difficult to detect [3]. There are some attempts on analyzing educational time series data [4], evaluating learners' emotional changes throughout learning process [5], but they didn't consider to make analysis on learners' cognitive level. Some work tried to do cognitive analysis of learning [6], but they didn't combine it with temporal data mining and consider using deep learning techniques.

An intelligent teaching environment helps educators to communicate with learners and be informed of recent states of learners. These technologies make traditional teaching and learning more accurate and intelligent. The quality of education relies more on data analysis than on the experience of educators. Learners are involved in drawing up their learning plans at the

same time. Georgia state university tracks students from arrival to graduation in three years and has made a total of 100,000 active interventions based on the risk alert provided by the system, which has increased the graduation rate of students from 48% to 54% [7]. In Oregon's Beaverton, students' drop-off records, absenteeism records and various demographic information are used to help students adapt to school life better [8].

EAnalyst¹ solves the problem that learners have a hard time figuring out their own knowledge proficiency because of deficient assessment methods and inadequate guidance. Combing domain knowledge with educational data mining and analysis, EAnalyst enables learners to know their knowledge state from the dashboard and provides remedial learning strategy. EAnalyst is an end-to-end system that has been tested on both elementary schools and secondary schools. Thus, the system is designed mainly for learners in the K12 period. The system has been used by part of students of those schools since 2014 and gets notable results in controlled experiments.

2. DATASETS

The data of learners are collected cautiously and critically. Different datasets lead to different outputs. Data of too large or too small granularity can be harmful to the analysis process.

The main component of data collected by EAnalyst ranges from pre-class quiz, post-class quiz, homework, unit-test and term-test. We refer every quiz, homework or test as a collection of series exercises. The former three are mainly about inspecting learners' short-term mastery level on concepts they just learned and the latter two on a larger concept coverage area. Exercises can be both online and offline. Educators use tools provided by the platform to select questions from question bank to form test papers. While offline exercises are commonly used for learners at a young age using the traditional paper test, online exercises are mainly taken on digital devices which can help collecting more information from question answering process such as time spent per question.

3. SYSTEM ARCHITECTURE

We describe EAnalyst architecture illustrated in Figure 1. EAnalyst is composed of preprocess module, analysis module, dashboard module and recommendation module. Preprocess module takes test papers and answer sheets as inputs and outputs structured data; analysis module takes structured data as input, outputs analysis results; dashboard module and recommendation module take analysis results as input then output visualized analysis results and recommendation list.

¹ study.hub.nercel.com/#/

* Corresponding Author

Tao Huang, Zhi Li, Hao Zhang, Huali Yang and Hekun Xie
"EAnalyst: Toward Understanding Large-scale Educational Data"
In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.)
2020, pp. 620 - 623

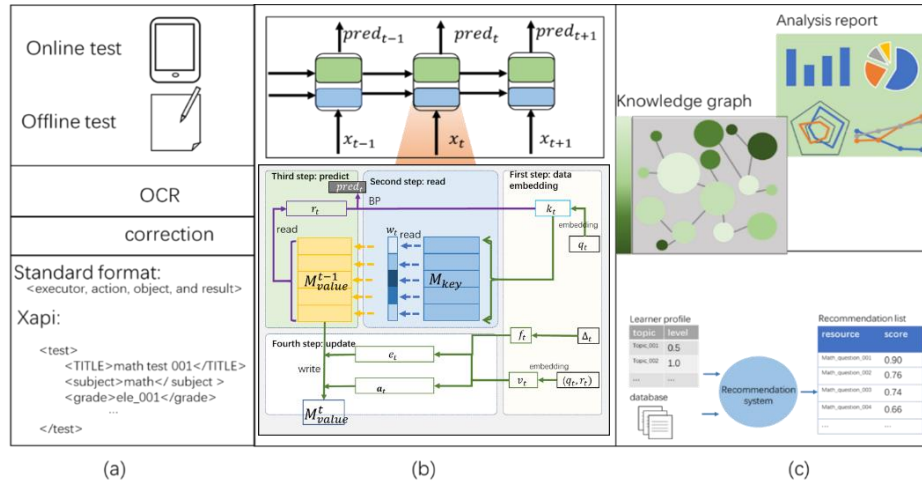


Figure 1. System architecture of EAnalyst: (a) Preprocess module. (b) Analysis module. (c) Dashboard module and recommendation module

3.1 Preprocess Module

Preprocess module uses optical character recognition (OCR) to transfer handwritten answers and correction marks to machine-encoded text. The module applies Transformer [9] which is one of natural language process (NLP) techniques to comprehensively learning question representation so that it can label questions with corresponding knowledge concepts. The response of learners to questions are recorded after being corrected by educators. The module then formalizes those heterogeneous educational data using the Experience API (Xapi), which makes the data readable for machine. Figure 1(a) illustrates EAnalyst's preprocess module.

3.2 Analysis Module

Learners interact with their coursework and generate sequences of learning process records. A sequence consists of multiple interaction record x_0, \dots, x_t . The task of this module can be seen as predicting learner's future performance x_{t+1} . The record x_t at time step t can be represented as $x_t = (q_t, a_t)$ where q_t is a question learner attempts at time step t and $a_t \in \{0, 1\}$ means learner's response (1 means correct and 0 means incorrect). Learning history is then analyzed by knowledge tracing model to reveal learners' learning status. From knowledge tracing prediction, educators can identify specific areas where learners need extra help. Educators can also analyze the data of the whole class to see their learning habit and adjust courses according to the feedback. Educators can even compare this information with that from other grades to determine which teaching methods are most effective.

The datasets that are used by knowledge tracing model are collected during the 2017-2019 school years. The datasets we conducted experiments on is on math subject, which has covered 652752 practice attempts of 3962 students on 4784 distinct questions. We filter learners who has fewer than three exercises to guarantee the reliability of knowledge tracing results since sequences that only contain one or two exercises barely contribute to tracing knowledge state of learners. We summarize some statistical features of two datasets in Table 1 and EAnalyst dataset distribution in Figure 2. For EAnalyst dataset, the average number of records per learner is 165. For EAnalyst dataset each learner interacts with more distinct questions than that in open dataset, which makes EAnalyst dataset more sparse.

Deep learning has made a huge success in tasks like image recognition, natural language processing (NLP), voice recognition and etc. Tasks which are good at handling sequential data use model like Long Short-Term Memory (LSTM) networks [10], a type of Recurrent Neural Networks (RNN), and get good results. Compared with models based on statistical graph like Bayesian Knowledge Tracing [11] and models based on matrix decomposition like Knowledge Proficiency Tracing [12], models based on deep learning, called Deep Knowledge Tracing (DKT) [13] are more flexible, which can be combined with effective mechanics so that they can make use of other information like content of questions and domain knowledge. DKT uses LSTM and its variation to cover previous learning records in a long time period to detect learners' knowledge state and memorize it in hidden vectors. This method has been combined with the attention mechanism to evaluate similarity among different question contents to improve prediction accuracy [14].

Table 1. Statistics of two datasets

Dataset Name	EAnalyst Dataset		Assistment2009 Dataset
Attribute of Dataset	Original	Pruned	Original
records	657573	652752	525534
learners	4285	3962	15931
questions	4788	4784	124

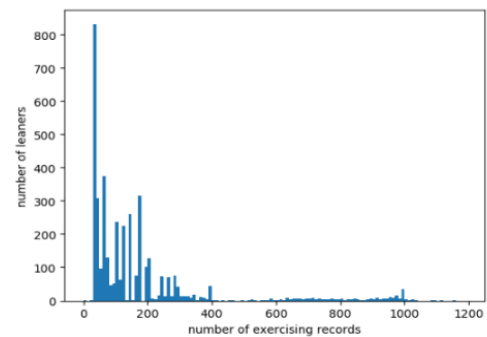


Figure 2. Distribution of EAnalyst Dataset on math subject.

Models like memory network [15] which has worked well in NLP field has also performed well at learning correlation between different questions. Model [16] using static key memory matrix to store question-concept relationships and dynamic value memory matrix to store and update concept-learner state relationships. This model performs well at knowledge tracing. We inherit the advantage of two memory matrices and apply convolution neural networks with some additional calculation to the reading process in the third step to reduce information loss in reading memory matrix process. We also consider the forgetting behavior of learners and add time interval of adjacent exercises to the updating step so that the model can simulate forgetting behavior. At first step, input data will be embedded. At second step, a question q_t is used to retrieve related concept position w_t in key matrix. At third step, position w_t is used in value matrix to query corresponding concept state. Finally, the concept state is used to predict learner's future performance on q_t . At fourth step, only related concept state will be updated in value matrix. The overall structure is illustrated in Figure 1(b).

We compare the prediction accuracy on both our dataset and public benchmark dataset—Assistment2009 [17]. Assistment is an online platform which teaches and assesses learners in elementary school mathematics. It is also the largest available public knowledge tracing dataset. We use Area Under a ROC Curve (AUC) to measure performance of the traditional model and deep learning model. AUC value ranges from 0.5 to 1 where the former value indicates the prediction result by random guessing and the latter represent precise prediction.

We set all sequences to be length of 150 and use -1 to pad short sequences to the expected length. The parameters are initialized randomly using Gaussian distribution. We set batch size for Assistment2009 dataset to 32 and that for EAnalyst dataset to 16 due to limitation of gpu memory. For momentum, it is set to be 0.9 and for norm clipping threshold to be 50.

The performance of different models is listed in in Table 2. The comparison results lead to findings that EAnalyst model can produce relative good result on Assistment2009 and better prediction results on EAnalyst dataset considering EAnalyst dataset are much sparser than Assistment2009. And Our model does not come into the problem of overfitting due to its complexity compared to DKT's LSTM network.

Table 2. Performance of different models on two datasets – EAnalyst dataset and Assistment2009 dataset (AUC)

Model	EAnalyst Dataset	Assistment2009 Dataset
Bayesian Knowledge Tracing	0.69	0.73
Variant of Bayesian Knowledge Tracing	0.75	0.82
Deep Knowledge Tracing on EAnalyst platform	0.85	0.86

3.3 Dashboard Module

Dashboard module is a visualization tool for learners displaying results of analysis on knowledge graph, which is illustrated in Figure 1(c) upper part. Educators and experts in education field construct the knowledge graph manually according to textbooks and their experience. Knowledge graph constructs a network of knowledge concepts, which are connected by lines with relevant knowledge concepts. The size of each concept is related to its

importance. The importance level is valued by corresponding syllabus. The more important a concept is, the bigger is a node. Color depth of a node indicate how a learner mastery a concept node. Each subject includes multiple knowledge graphs divided by school year while some concepts can appear in one or more graphs. Knowledge graph is a precondition of accurate analysis of learners' overall cognitive levels, knowledge state and appropriate learning path recommendation. A learner and his or her educator can locate weak spots easily. And having a big picture of one's knowledge state helps the learner to carry out the following remedial activities.

Analysis report giving a more detailed description of a learner's learning report. History of exercises will be evaluated in a statistical point of view. Different types of charts such as histogram, pie chart, radar chart and line chart. These charts can well represent changes in learning indicator of learners over time, break out learners of a class by percentage of accuracy they have got, show distribution of a learner's overall quality and give a rough comparison between the learner and the average level of his or her class and grade. Figure 3 gives a partial screenshot of a learner's dashboard in elementary school mathematics.

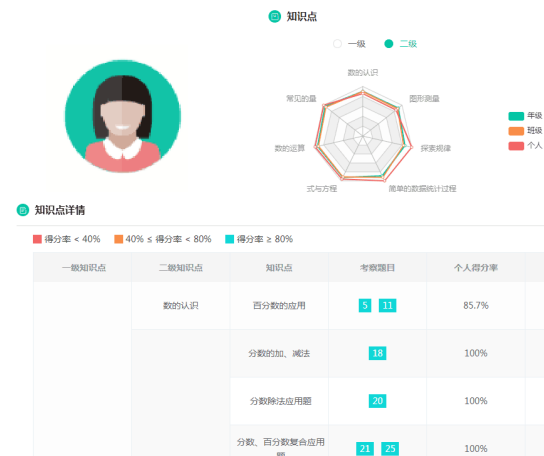


Figure 3. concepts mastery level in a radar chart and statistical report

Analysis report giving a more detailed description of a learner's learning report. History of exercises will be evaluated in a statistical point of view. Histogram represents change in learning indicator like accuracy over time. Pie chart breaks out learners of a class by percentage of accuracy they have got. Radar chart shows distribution of a learner's overall quality. Line chart gives a rough comparison between the learner and the average level of his or her class and grade.

Dashboard contains statistical reports generated from analysis module and knowledge graph presenting learner's knowledge proficiency. The report displays learner's test results, test analysis. The circle in the graph represents separate entities. The importance of the entity is distinguished by size, and the depth of color indicates the learners' mastery level of each entity. The line between two circles displays relation existing between two corresponding entities. Dashboard works as an effective tool to promote learners to define and achieve goals.

3.4 Recommendation Module

Recommendation module mines learner features and course features, uses learners' rating of learning materials as supervised labels to filter recommendation materials like reading material, exercises, notes and outstanding answers from learning partners. We form a learner-course feature vector matrix by combining learners' behavior data with attributes data from learners and courses. This module first uses extraction capabilities of deep belief networks (DBN) to collect features from learner-course matrix to represent learners' preference. This feature extraction part is composed of bottom-up unsupervised pretraining using layers of restricted Boltzmann machine (RBM) and top-down supervised parameter fine-tuning using Backpropagation (BP) in the last level of the DBN. The trained DBNs from unsupervised part and corresponding rating score labels are used as inputs to the BP supervised part [18]. Then the recommendation model can be used to rating learning materials with scores. Materials with scores are ranked and those with higher scores are recommended to learners. The process is illustrated in Figure 1(c) lower part. This recommendation list will be updated dynamically according to newly generated learning tracks to match learners' changing needs.

4. CONCLUSION

We present EAnalyst, a learner's assistant developed by applying deep learning techniques for large-scale educational data mining and analysis. The system takes temporal data analysis aligned with knowledge graph, presents learners with multidimensional analytical reports, and recommending learning paths by offering relative learning materials. In the future, we intend to solve the "cold start" problem of learners' performance evaluation process and improve the analysis model by adding question content so that the deep relation between questions and learners' state can be exploited.

5. ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China under grants No. 61977033.

6. REFERENCES

- [1] Wang Y, Liao H C. 2010. Data mining for adaptive learning in a TESL-based e-learning system. *Expert Systems with Applications*, 2011,38(6): 6480-6485. DOI=<https://doi.org/10.1016/j.eswa.2010.11.098>
- [2] Brown N C C, Kölling M, McCall D, et al. 2014. Blackbox: a large scale repository of novice programmers' activity. *Proceedings of the 45th ACM technical symposium on Computer science education*. ACM, 2014: 223-228. DOI=<https://doi.org/10.1145/2538862.2538924>
- [3] Allevato A, Thornton M, Edwards S, et al. 2008. Mining data from an automated grading and testing system by adding rich reporting capabilities. *Educational Data Mining 2008*. 2008.
- [4] Baker R S, Inventado P S. 2014. Educational data mining and learning analytics. *Learning analytics*. Springer, New York, NY, 2014: 61-75. DOI=https://doi.org/10.1007/978-1-4614-3305-7_4
- [5] Le N T, Boyer K E, Chaudry B, et al. 2013. The First Workshop on AI-supported Education for Computer Science (AIEDCS). *International Conference on Artificial Intelligence in Education*. Springer, Berlin, Heidelberg, 2013: 947-948. DOI=https://doi.org/10.1007/978-3-642-39112-5_159
- [6] Schunk D H, Greene J A. 2017. Handbook of self-regulation of learning and performance. Routledge, 2017.
- [7] Executive Office of the President, Munoz C, Director D P C, et al. 2016. Big data: A report on algorithmic systems, opportunity, and civil rights. *Executive Office of the President*, 2016.
- [8] West D M. 2014. Big data for education: Data mining, data analytics, and web dashboards. *Governance studies at Brookings*, 2012, 4(1).
- [9] Vaswani A, Shazeer N, Parmar N, et al. 2017. Attention is all you need. *Advances in neural information processing systems*. 2017: 5998-6008.
- [10] Hochreiter S, Schmidhuber J. 1997. Long short-term memory. *Neural computation*, 1997, 9(8): 1735-1780. DOI=<https://doi.org/10.1162/neco.1997.9.8.1735>
- [11] Yudelson M V, Koedinger K R, Gordon G J. 2013. Individualized bayesian knowledge tracing models. *International conference on artificial intelligence in education*. Springer, Berlin, Heidelberg, 2013: 171-180. DOI=https://doi.org/10.1007/978-3-642-39112-5_18
- [12] Chen Y, Liu Q, Huang Z, et al. 2017. Tracking knowledge proficiency of students with educational priors. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017: 989-998. DOI=<https://doi.org/10.1145/3132847.3132929>
- [13] Piech C, Bassen J, Huang J, et al. 2015. Deep knowledge tracing. *Advances in neural information processing systems*. 2015: 505-513.
- [14] Su Y, Liu Q, Liu Q, et al. 2018. Exercise-enhanced sequential modeling for student performance prediction. *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.
- [15] J. Weston, S. Chopra, and A. Bordes. 2015. Memory networks. *International Conference on Learning Representations*, 2015.
- [16] Zhang J, Shi X, King I, et al. 2017. Dynamic key-value memory networks for knowledge tracing. *Proceedings of the 26th international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017: 765-774. DOI=<https://doi.org/10.1145/3038912.3052580>
- [17] Feng M, Heffernan N, Koedinger K. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction*, 2009, 19(3): 243-266. DOI=<https://doi.org/10.1007/s11257-009-9063-7>
- [18] Zhang H, Huang T, Lv Z, et al. 2019. MOOCRC: A Highly Accurate Resource Recommendation Model for Use in MOOC Environments. *Mobile Networks and Applications*, 2019, 24(1): 34-46. DOI=<https://doi.org/10.1007/s11036-018-1131-y>

How we talk about math: Leveraging naturalistic datasets to define the discourse of math in contrast to other domains

Rachel Jansen
Department of Psychology
University of California, Berkeley
Berkeley, CA 94720 USA
racheljansen@berkeley.edu

Ruthe Foushee
Department of Psychology
University of California, Berkeley
Berkeley, CA 94720 USA
foushee@berkeley.edu

ABSTRACT

How do people talk about math? What point are we making when we contrast math with other topics? In studies of school performance, attitudes, and stereotypical beliefs, math is most frequently compared to language abilities and occasionally artistic qualities. Most studies about these topics administer assessments and closed-form surveys to make sense of how math ability or beliefs are different from similar constructs in other educational domains. In an analysis of Google search terms using Google Trends, “math” occurs in search queries far more frequently than “language” or “art” and —unlike searches about the other topics—the prevalence of “math”-related searches shifts in conjunction with the academic year. This project’s goals are to (1) sample from diverse naturalistic text-based datasets to expose how math is referred to in non-experimental settings and (2) identify similarities and differences between math and the domains most frequently used as contrasts. We perform computational analyses on text derived from naturalistic sources written across a variety of different registers, from a journalistic source (NY Times) and a social media website (Twitter) to referential sources containing basic definitions (Merriam-Webster) and more informal descriptions (Urban Dictionary). We see that, across data sources, queries related to “math” refer more frequently to education-related themes and incorporate more disparaging terminology compared to content related to “language” or “art.” This project is a first step in demonstrating that this methodology can aid in exploring more realistic discourse surrounding math and domains of comparison. This can inform and empower future researchers and practitioners interested in changing the discussion around math.

Keywords

math, language, art, naturalistic data, big data, NLP

Rachel Jansen and Ruthe Foushee "How we talk about math: Leveraging naturalistic datasets to define the discourse of math in contrast to other domains" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 624 - 628

Math is frequently discussed in a derogatory way: “Everybody hates math” is a major trope in popular media, with hundreds of instances across television sitcoms, comics, and movies.¹ In addition, when researchers investigate attitudes about math, they tend to focus their efforts on the study of math *anxiety* [2, 6], as opposed to positive feelings. These ways of portraying math, both in the media and in research and education appear unique to math. When researchers measure stereotypical beliefs [4] or attitudes about another subject [8], they are typically included as a contrast to math.

In public discourse, this differential treatment of math compared to other domains is perpetuated by *neuromyths*, or false ideas about the brain, such as the idea that “some of us are ‘left-brained’ and some are ‘right-brained’ and this helps explain differences in how we learn” [12]. Implying a natural contrast between math ability and art or language ability does a disservice to students, current and former: it encourages the belief that if we are “good” at one thing, we cannot be “good” at another. Beliefs about innate brilliance further amplify the folk distinction between math and art. Math is perceived as requiring the most brilliance out of any STEM discipline, and significantly more than all art and language-related fields, including English Literature, Art History, Linguistics, and Music Composition [10]. Such essentialist beliefs about domain-specific ability are bolstered by parallel sex disparities, with more “brilliant” fields like math including significantly fewer women.

In this project, we capitalized on naturally occurring data where people discuss math, and compare parallel discourse about other domains. We specifically analyze communications related to language and art, as these are frequently compared to math. We detail our rationale for each comparison domain in the next section.

0.1 Comparison domains

In research contexts, language very frequently serves as a comparison domain for math. Math is compared to language—and most often reading and writing skills—in research on ability [7], stereotyped beliefs [4], theories of intelligence [8]. [14] contrasts STEM performance (math and science) with non-STEM (language, humanities, and social science) and

¹<https://tvtropes.org/pmwiki/pmwiki.php/Quotes/EverybodyHatesMathematics>

finds no actual performance difference nor evidence for gender differences in variability in academic grades.

On the other hand, art is much less studied in relation to math, in part because it is far more difficult to design art assessments than math or language assessments, and judging art ability is perceived as subjective. Some work has contrasted creative ability in math and art [9] and explored gender differences in stereotypical beliefs across the two areas [16]. More frequently, when conducting research on a task that involves some artistic expression in research about math ability, the idea of “art” goes unmentioned, as with drawing [5] or any study of spatial ability [1].

In studies comparing math ability or perceptions to parallel constructs in another domain, no justification is given for the choice of alternative domain. It may seem obvious to us that reading ability is the most direct contrast to math ability, but this is precisely what we are interested in accessing in this project. The assumptions about concepts and distinctions among them present themselves in our communications. Therefore, investigating naturally occurring data may provide us with the justification we need for domain comparison choices across different contexts. We assert that though art is only sporadically studied in relation to math, there are many reasons why it may serve researchers as an appropriate foil. For example, math is a required course throughout schooling and necessary for attending college, while under budget shortages, art classes are the first to be cut from curricula. However, mathematicians regularly enjoy drawing comparisons between artistic and mathematical abilities [11, 3]. This perceived distinction in the relative utility of math and art, paired with experts’ regular likening of the two suggests an interesting avenue of future work.

0.2 Measuring math talk

Human attitudes are typically explored via closed-form surveys. But sampling bias as well as the wording of the questions can impact responses. We propose using naturally occurring datasets to supplement existing research about math attitudes and as a guide for developing new theories and experimental paradigms [15].

The goals of this paper are 1) to source data from non-experimental contexts to examine naturalistic discourse surrounding math and its comparison domains and 2) to identify how math is discussed that may be distinct from related domains in similar contexts. We hope to make an empirical case for comparing math to specific domains: why and when do we measure math against language or art, and what might be the appropriate choice based on how people represent these domains? We locate several sources of communication spanning a range of genres (e.g., journalistic, social media, and references), and registers (from more formal to informal writing styles).

1. METHODS

We identified a variety of online sources with freely accessible APIs (Application Programming Interfaces). We first used the Google Trends and English Lexicon Project as measures of frequency of term usage. Next we collected a selection of articles from the New York Times, tweets from Twitter, and definitions from the Merriam-Webster dictionary and

Urban Dictionary that related to the search terms “math,” “language,” and “art.”

1.1 Data sources

Though Google does not provide access to search history data, the company built an online interface, Google Trends,² for observing both fluctuations of searches for specific keywords or topics over time and across locations [17]. We focused our observations exclusively on the US. As another overview of frequency of specific terms, we used the English Lexicon Project (ELP).³ These sources provide a very general sense of how these topics are thought about differently. We next explore actual word usage in multiple other sources, namely the New York Times, Twitter, and two different online references (Merriam-Webster and Urban Dictionary). Two of these may be seen as relatively objective (Times and Merriam-Webster), though a computational analysis of word usage will show whether this is truly the case.

We used the “Article Search API” from the New York Times (NYT)⁴ to collect all hits that include the terms “math,” “language,” and “art.” The NYT API provides the headline, keywords, date, word count, and lead paragraph for all articles that come up for a specific search term. The NYT Article Search API yielded 441,773 searches for “math”, 367,707 for “language” and 1,276,036 for “art.” We sample approximately 2,000 results for each search term.

Twitter is the only social media company that offers easy access to their data, in part because posts are all expected to be public anyway.⁵ In order to align results with the data we obtained from the NY Times, we used the twitter package for Python⁶ to load 2,000 tweets per search term.

Merriam-Webster additionally offers easy access to their definitions.⁷ Preliminary data mining returned just the definitions for each term of interest, but it is meaningful that “math” has three definitions, while “language” and “art” each have ten. From Urban Dictionary, we downloaded all existing results for each term, which was 856 for math, 262 for language, and 876 for art (see Table 1 for total documents used in analyses for each term and each data source).

1.2 Text Analyses

With each data source, we created Naïve Bayes classifiers to contrast word usage for documents about math, language, and art.⁸ Prior to text analyses, we ran a series of standard text pre-processing techniques: a) removing stopwords b) removing punctuation and c) reducing words to their roots (stemming and lemmatizing). To test the accuracy of each classifier, we shuffle the data and separate it into a training set consisting of 80% of the data and a test set comprising the remaining 20%. We train the classifier on the training

²<https://trends.google.com/trends/?geo=US>

³<https://ellexicon.wustl.edu/>

⁴<https://developer.nytimes.com/>

⁵<https://developer.twitter.com/>

⁶<https://pypi.org/project/twitter/>

⁷<https://dictionaryapi.com/>

⁸We employ the NaiveBayesClassifier function from Python’s Natural Language Toolkit (nltk version 3.2.2) package <https://www.nltk.org/>

set, then report the classifier’s accuracy predicting responses on the test set, alongside a subset of informative features (words that are more common for one specific subgroup).

2. RESULTS

2.1 Google and ELP

From Fig 1, created from data generated in Google Trends to compare searches pertaining to the topics “math,” “language,” and “art,”⁹ it is clear that, compared to the other topics, math is generally searched for at a higher rate, though takes steep dives in the summer months when school is no longer in session. This suggests that the term “math” is much more associated with education than other topics, and idea we explore in more detail in the other data sources. Contrary to Google search results, an analysis of term frequency in the English Lexicon Project reveals “math” to be significantly lower frequency (18,404) than terms relating to “language” (97,874) or “art” (62,513). According to this source, “math,” “language” and “art” are estimated to be acquired at similar ages (5.56, 6.79, and 6.21 years, respectively), but “language” is rated as notably more abstract (that is, less concrete: language: 2.35, math: 3.15, art: 4.17).

	MATH	LANGUAGE	ART	TOTAL
NY TIMES	1,541	1,946	1,835	5,322
TWITTER	2,000	2,000	2,000	6,000
M. WEBSTER	3	10	10	23
URBAN DICT.	856	262	876	1,994

Table 1: Number of documents for each corpus.

2.2 Journalistic source

We excluded a set of “math” searches to ensure that the results would not be overly skewed. Specifically, 270 hits contained a daily math challenge and the lead paragraph began with “Test your math skills with today’s question” and an additional 40 started with “Our weekly math problems are written by teachers at Math for America.” For “art,” we excluded 157 whose lead paragraphs began with “Our guide to new art shows and some that will be closing soon.” There did not appear to be anything so consistent for searches relate to “language.” Search results with blank lead paragraphs (159 math; 18 art; 64 language) were excluded from our training data. First, we analyzed the distribution of keywords. Of the 1541 math queries, 323 contained “school” (21%) compared to 81 of 1946 language queries (4%) and 14 of the 1835 art queries (0.8%). There was a similar pattern for the keyword “test,” included in 149 math queries, compared to 6 and 1 for language and art, respectively.

We next looked at the text from each lead paragraph. We used all three sets of nonempty lead paragraphs for each topic to construct the classifier which included a total of 5,322 texts (1541 math; 1946 language; 1835 art). After removing all terms with roots “math,” “language,” or “art,” the classifier achieved 70% accuracy on the test set. The most informative features for math included “test,” “score,” “grader,” “improv,” “educ,” “competit,” and “teacher” (e.g., “Growing up, I thought math class was something to be

⁹<https://trends.google.com/trends/explore?date=all&geo=US&q=%2Fm%2F04rjg,%2Fm%2F04g7d,%2Fm%2F0jjw>

endured, not enjoyed. I disliked memorizing formulas and taking tests, all for the dull goal of getting a good grade”). The most predictive terms for an art-related hit contained “galleri,” “sculptur,” “paint,” and “noteworthy.” For language-related queries, “speak,” “translat,” “dictionari,” and “writer” were most informative. Though there are no apparent emotive terms, math arises much more frequently in documents related to school than does art or language in this context.

2.3 Social media

The average word count for tweets corresponding to each term was approximately equal (18 for math, 19 for language, and 18 for art), likely due to platform word count restrictions. In our classifier (accuracy: 62%), informative features for tweets about math included words very similar to those from the NYT, such as “test,” “fail,” “wrong,” “class,” and “science.” For language, we saw “tiktok,” “english,” “speak,” “video,” and “utter.” The set of most informative features for art contained “draw,” “style,” “anim,” “design,” and “cute.” Here, we see many domain-specific similarities to the NYT data, but with the addition of terms conveying negative emotions related to math, such as “wrong,” “fail,” and “hard,” which might speak to the greater subjectivity of the text source. The language- and art-related searches also appear to encompass more popular culture references. There was an interesting pattern of math being more ubiquitous in tweets relating to current events such as the election: ‘berni,’ ‘vote,’ and ‘warren’ (e.g., “Math says that Warren has a path”) and the coronavirus outbreak: “million,” (e.g., “It is simple math. The flu infects millions a year”).

2.4 Reference materials

The Merriam-Webster dictionary produced few results, but the primary definitions themselves serve as a baseline set of relevant objective terms. Mathematics is defined as “the science of numbers and their operations,” language as “the words, their pronunciation, and the methods of combining them used and understood by a community,” and art as “skill acquired by experience, study, or observation.” By their very definitions, math is a science (rather than an art) and art is not said to require innate ability.

The Urban Dictionary API yielded 856 definitions of “math,” 262 of “language,” and 876 of “art.” The mean length of the math definitions was 36 words, 49 for language, and 58 for art (similar to the NYT results). Because this corpus was not evenly distributed across topics, we ran separate classifiers between each pair of topics, rather than over all 1,994 total definitions. For the math/art classifier ($n = 1732$), accuracy on the test set was 79% and the most informative words for math definitions were almost all negative: “abus,” “number,” “mental,” “stress,” “tortur,” and “bore.” For art on the other hand, informative features included “style,” “emot,” “draw,” “amaz,” “visual,” and “color.” The classifier comparing math to language yielded an accuracy of 85%¹⁰ and primarily identified words that were informative of the language texts, as they represented a smaller proportion of our dataset. These included “speak,” “talk,” and “special,” while terms more indicative of a math entry were “abuse,” “mental,” “human,” and “bore.”

¹⁰Chance would be 77% because 77% of definitions are math ones.

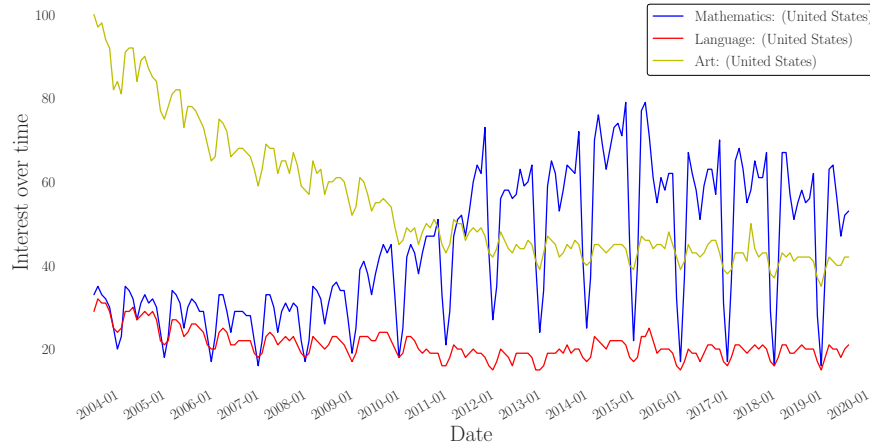


Figure 1: Google trends patterns of searching for the topics “mathematics,” “language,” and “art.”

Finally, the classifier comparing art and language arrived at an accuracy of 85% and words from art-specific definitions included “music,” “best,” and “style,” while definitions of language included terms “sign,” “french,” “wrong,” and “number,” which potentially likens math to language more than to art. Analyses of the example sentences included for each definition from Urban Dictionary produced comparable results, with notably more derogatory and profane terminology used to describe math than the comparison domains.

3. DISCUSSION

In a preliminary analysis of naturally occurring data sources, we have observed that math is more frequently written about and discussed in relation to education compared to language and art. In contexts where valenced language use is common (Twitter and Urban Dictionary), math is discussed using notably more unflattering terminology. Each data source we explored yielded different frequencies at which the three chosen topics were mentioned. Our Google Trends analysis revealed that math is searched for more frequently than language or art. However, though the NYT provided fewer “language” articles than “math” ones, there were more than double the number of hits related to “art” (owing to “arts and leisure” having its own section in the newspaper). In the references, “math” had fewer entries in Merriam-Webster compared to language and to art, but in Urban Dictionary, there were a comparable number of entries for “math” and “art,” and this was more than triple the number of “language” entries.¹¹ Based solely on these simple search counts, we can identify important differences in how these topics are thought about: “math” appears to be defined more narrowly than the other domains (based on Merriam-Webster definition counts and shorter text lengths in the NYT and Urban Dictionary data) while emotions surrounding “math” and “art” are stronger than for “language” (based on the relative number of Urban Dictionary results). “Math” is also much more associated with education, a claim supported by the keywords from the NYT, our classifiers’ informative fea-

tures from the NYT, Twitter, and Urban Dictionary data, and from the cyclical nature of Google searches for “math.”

4. FUTURE DIRECTIONS

This set of analyses only scratches the surface of what is possible with this methodology. We have many plans for further research, namely to conduct additional analyses on the data presented here, gather more data through clouds of novel search terms, and explore other naturally occurring data sources. First, to expand the findings from the data already gathered, we will conduct text-based sentiment analyses to search for systematic differences in overall valence associated with each term, and perform topic modeling over each set of documents. Second, “language” and “art” are only two of many possible domains to compare to math, alternatives to which we will pursue in future work. We aim to use the work done so far to refine our terms to determine what comparisons are more useful for different contexts. Finally, we intend to search deeper and with more specific intentions within the sources we have scrutinized thus far as well as among other potential sources of data.

5. CONCLUSION

Using large-scale datasets of naturally occurring text, this work presents a preliminary exploration of how math is discussed, compared to its most frequent comparison domains. Our data confirm that math is generally spoken about in a manner that is both more limited (e.g., to educational contexts) and more negatively valenced. Previous work has shown that familiarity with an idea increases belief in that idea [18], which means that the restricted and unflattering ways in which we generally discuss math may progressively degrade public opinion about the topic. If—through the media and other sources—speakers continue to hear (or read) about math as a narrowly defined concept associated with negative emotions, this perception will continue to thrive, and be unwittingly transmitted to future generations [13]. Thus, this work also serves as a plea to limit unnecessary disparaging reference to math in mass communication.

¹¹We were not able to acquire total hit numbers from Twitter.

6. REFERENCES

- [1] D. Barner, G. Alvarez, J. Sullivan, N. Brooks, M. Srinivasan, and M. C. Frank. Learning mathematics in a visuospatial format: A randomized, controlled trial of mental abacus instruction. *Child Development*, 87(4):1146–1158, 2016.
- [2] E. Carey, F. Hill, A. Devine, and D. Szűcs. The chicken or the egg? The direction of the relationship between mathematics anxiety and mathematics performance. *Frontiers in Psychology*, 6:1987, 2016.
- [3] G. J. Chaitin. *Conversations with a Mathematician: Math, Art, Science and the Limits of Reason*. Springer Science & Business Media, 2002.
- [4] E. K. Chestnut and E. M. Markman. “Girls are as good as boys at math” implies that boys are probably better: A study of expressions of gender equality. *Cognitive science*, 42(7):2229–2249, 2018.
- [5] J. E. Fan. Drawing to learn: How producing graphical representations enhances scientific thinking. *Translational Issues in Psychological Science*, 1(2):170, 2015.
- [6] A. E. Foley, J. B. Herts, F. Borgonovi, S. Guerriero, S. C. Levine, and S. L. Beilock. The math anxiety-performance link: A global phenomenon. *Current Directions in Psychological Science*, 26(1):52–58, 2017.
- [7] L. Guiso, F. Monte, P. Sapienza, and L. Zingales. Culture, gender, and math. *Science*, 320(5880):1164–1165, 2008.
- [8] E. A. Gunderson, N. Hamdan, N. S. Sorhagen, and A. P. D’Esterre. Who needs innate ability to succeed in math and literacy? Academic-domain-specific theories of intelligence about peers versus adults. *Developmental psychology*, 53(6):1188, 2017.
- [9] K.-N. Jeon, S. M. Moon, and B. French. Differential effects of divergent thinking, domain knowledge, and interest on creative performance in art and math. *Creativity Research Journal*, 23(1):60–71, 2011.
- [10] S. J. Leslie, A. Cimpian, M. Meyer, and E. Freeland. Expectations of brilliance underlie gender distributions across academic disciplines. *Science*, 347(6219):262–265, 2015.
- [11] P. Lockhart. *A mathematician’s lament: How school cheats us out of our most fascinating and imaginative art form*. Bellevue literary press, 2009.
- [12] K. Macdonald, L. Germine, A. Anderson, J. Christodoulou, and L. M. McGrath. Dispelling the myth: Training in education or neuroscience decreases but does not eliminate beliefs in neuromyths. *Frontiers in psychology*, 8:1314, 2017.
- [13] E. A. Maloney, G. Ramirez, E. A. Gunderson, S. C. Levine, and S. L. Beilock. Intergenerational effects of parents’ math anxiety on children’s math achievement and anxiety. *Psychological Science*, 26(9):1480–1488, 2015.
- [14] R. E. O’Dea, M. Lagisz, M. D. Jennions, and S. Nakagawa. Gender differences in individual variation in academic grades fail to fit expected patterns for stem. *Nature communications*, 9(1):1–8, 2018.
- [15] A. Paxton and T. L. Griffiths. Finding the traces of behavioral and cognitive processes in big data and naturally occurring datasets. *Behavior Research Methods*, 49(5):1630–1638, 2017.
- [16] J. R. Steele and N. Ambady. “Math is hard!” The effect of gender priming on women’s attitudes. *Journal of Experimental Social Psychology*, 42(4):428–436, 2006.
- [17] S. Stephens-Davidowitz and H. Varian. A hands-on guide to google data. 2014.
- [18] W.-C. Wang, N. M. Brashier, E. A. Wing, E. J. Marsh, and R. Cabeza. Knowledge supports memory retrieval through familiarity, not recollection. *Neuropsychologia*, 113:14–21, 2018.

Predicting Student Dropout by Mining Advisor Notes

J.D Jayaraman

New Jersey City University and Teachers College, Columbia University

jjayaraman@njcu.edu

ABSTRACT

More Americans are attending college than ever before but almost half of them do not complete college. Thus, early detection of students at risk of dropping out of college is of paramount importance. This study describes a novel attempt at using notes made by student advisors to predict student dropout. We use a Natural Language Processing (NLP) technique called sentiment analysis to analyze unstructured textual data to extract the positive or negative sentiment contained in the advisor's notes. We then use the sentiment extracted from the notes as features to train a random forest model to predict student dropout. We achieve 73% accuracy in predicting student dropout. Thus, our study demonstrates the value of unstructured data held in institutional databases for identifying at-risk students.

Keywords

Dropout prediction, Sentiment analysis, Machine learning models, At-risk students, Natural language processing, Text mining

1. INTRODUCTION

Student retention is a major challenge at American universities with the average 6 year graduation rate hovering around 59% [12]. Graduation rates vary with institutional selectivity [19]; the situation being particularly grave at institutions with open admission policies where the 6 year average graduation rate is a meager 32% [12]. Low retention rates not only impact the financial well-being of individuals but the economy as a whole, since it is a well-established fact that income level rises with a college degree. Median income levels for young adults with a bachelor's degree are 64% higher than those with only a high school diploma [12]. Low retention rates also adversely affect the reputation of the educational institution and could lead to potential loss of funding and inability to compete for quality students. Thus, improving student retention is of paramount importance at institutions of higher education.

A critical factor in increasing student retention is the ability to accurately identify at-risk students, so that relevant interventions can be provided. Much of the prior research has been devoted to modeling the factors that impact student retention using traditional statistical methods. But, machine learning and data mining techniques have started becoming actively employed in student retention research in the recent past. Most research articles, though, have been focused on using structured data, such as GPA, SAT scores etc., that are readily available in institutional databases. To

the best of our knowledge, as of this writing, there is no literature that tries to use unstructured data (e.g. free form text, images etc.) in predicting student dropout. Roughly 80% of the data generated in the world today is unstructured. Large amounts of unstructured data are generated by universities and colleges. Examples include advisor notes, discussion forum postings, online chats, emails etc. This is a treasure trove of information that has not been adequately exploited to help predict student dropout.

This paper describes a novel approach to predicting college student dropout using the information contained in free form notes recorded by student advisors on a student advising platform (e.g. EAB). We use Natural Language Processing (NLP) techniques to unearth the information contained in these advisor notes and use it to predict student dropout. To the best of the author's knowledge this study is one of the first to employ NLP techniques to predict student dropout. Thus, our study contributes to the literature by introducing an additional novel approach to predicting student dropout by using NLP techniques to analyze unstructured textual data in the form of advisor notes.

2. LITERATURE REVIEW

Research on student attrition has traditionally been based on surveying student cohorts and following them to assess dropout. These surveys contributed to the building of theoretical models of student retention, the most famous of them being the Tinto model [16]. Survey based research have been criticized for being too specific to an institution and hence not generalizable [1]. Also, these large scale surveys are not cost-effective to conduct. An alternative to survey based research is to use the data that most higher education institutions routinely collect about their students. This type of research based on institutional databases has been shown to be comparable to survey based research [2].

Prior research has also been mostly focused on identifying various factors that impact student dropout. Tinto [17] highlights academic difficulty, adjustment problems, lack of clear academic goals, lack of commitment, inability to integrate with the college community, uncertainty, incongruence and isolation as factors involved in student dropout. Tinto's theory of student integration posits that past and current academic success are crucial factors in determining student attrition and many studies have found high school GPA and SAT scores to have a strong effect on student retention [13]. Declaration of major and number of credit hours taken during the first semester have been used as proxies for institutional and goal commitment and have been found to be significant predictors of student attrition [1]. There have been many studies that have investigated the effect of financial aid on student retention [8, 9, 14]. These studies found that the type of financial aid that the student received had an impact on student retention.

Students receiving aid based on academic achievement had higher retention rates, while student loans had a negative effect on retention. Also, if students lost a scholarship or grant due to poor grades, it had a negative impact on retention. Thus, as evidenced above, almost all the studies have focused on factors that are part

J.D Jayaraman "Predicting Student Dropout by Mining Advisor Notes" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 629 - 632

of structured data collected by educational institutions. Factors such as emotions and sentiments that are embedded in unstructured data, have not been considered much in the literature.

Research on using machine learning techniques to predict student attrition is still in its infancy. Delen [6] used a dataset consisting of 39 variables such as SAT score, high school GPA, hours registered, hours earned etc. and several machine learning methods such as support vector machines and neural networks to model freshmen student attrition and found that support vector machines performed best, reaching a prediction accuracy close to 80%. Thammasiri, Delen, Meesad and Kasap [15] used data and techniques similar to Delen [6] to predict whether students would enroll for the second term. Lauria, Baron, Devireddy, Sundararaju, and Jayaprakash [10] used demographic and course related data to show that support vector machines performed better than decision trees at predicting at-risk students. Thus, almost all the research on student dropout prediction using machine learning and statistical techniques has focused on using structured data.

While there is not much literature using NLP techniques and unstructured data in predicting college student dropout, there is some recent literature in the related area of predicting student completion in Massive Open Online Courses (MOOCs). The most common NLP technique employed in these studies is sentiment analysis, which examines language in discussion forums and assignments to detect positive or negative emotion words and words that convey motivation, engagement etc. Wen, Yang, and Rose [18] examined students' opinion towards the course based on a sentiment analysis of discussion forum posts and used these opinions to predict course completion. Wen et al. [18] found that students who used words related to motivation were more likely to complete the course. Crossley, Paquette, Dascalu, McNamara, and Baker [4] used NLP techniques on MOOC forum posts and found that lexical sophistication, writing quality were predictive of student completion. Our study uses similar approaches to the literature described above but applies sentiment analysis to free form notes entered into an advisement system by the student's advisor, in order to predict student dropout.

3. METHODOLOGY

3.1 Data

The data consists of 19,562 notes entered over a period of four years (2015 - 2018) for 7343 undergraduate students at an urban university in the North Eastern United States which caters to a largely minority population. These notes are made by the student's advisor after each meeting with the student and are keyed into the student advisement system. These notes are free form and do not have any structure to them. Students typically meet with the advisor multiple times a semester to discuss enrollment, progress and any other issues. The notes the advisor makes documents the meeting in a reasonable amount of detail. Thus the notes are rich with information on any issues and difficulties students might be facing not only with respect to their academics but also with respect to their social and family life. We also compiled data on whether a student dropped out or not (a binary indicator variable). A student was considered to have dropped out if he or she did not enroll in any semester following the last semester of enrollment. Based on this definition we constructed a binary indicator variable to indicate whether a student has dropped out or not.

3.2 Analysis

3.2.1 Sentiment Analysis

Sentiment analysis is a NLP technique that attempts to categorize the emotions and sentiments in a block of text. Most sentiment analysis tools will categorize the sentiment as positive, negative or neutral and also provide indexes for affective states such as anger, sadness, happiness, etc. Sentiment analysis has been widely used to mine emotions from social media posts and has been effective in identifying depression, anxiety and other emotions [15].

There are two main approaches to extracting sentiment from text. The lexicon based approach uses a dictionary of words annotated with their sentiment polarities, while the text classification approach involves building classifiers from labelled instances of texts. Lexicon based approaches work well when there is insufficient human classified data or when human classification is time consuming and expensive. We use the lexicon based approach in this study as it would be very time consuming to hand classify the sentiment in the advisor notes to create a large enough training dataset. There are several sentiment lexicons available. We use a popular lexicon called the Bing lexicon [11] which consists of 6800 words, 2000 of which are positive and 4800 are negative. We also constructed a custom lexicon of 100 sentiment words relevant to the student retention domain and combined it with the Bing sentiment lexicon.

We preprocessed the data by removing stop words, punctuations, numbers, white spaces and other words such as will, student, etc. that would not be pertinent to conveying sentiment. The sentiment analysis was done on the preprocessed data. The output of the sentiment analysis is a list of words in each note tagged with a sentiment (positive or negative).

3.2.2 Imbalance

Data is said to be imbalanced if the number of instances in one class significantly outnumbers the number of instances in other classes. Since the number of dropouts is much smaller when compared to the number that don't dropout, student retention data sets are typically imbalanced. If the data is imbalanced the standard classifiers have a bias towards the larger majority class. One approach to correcting this imbalance is to preprocess the data in order to balance it out and then build the model. This approach uses various techniques to either oversample the minority class or undersample the majority class. Random oversampling attempts to balance the data by randomly sampling from the minority class and adding them to the training data set while random undersampling attempts to balance the data by removing data instances from the majority class. Undersampling has been shown to perform better than oversampling in some cases [7]. Synthetic Minority Oversampling Technique (SMOTE) is a popular and robust technique that uses a combination of oversampling the minority class and undersampling the majority class which results in better classifier performance than just oversampling or undersampling [3]. Our study uses SMOTE to correct the imbalance.

3.2.3 Classification

From the output of the sentiment analysis we computed the number of positive sentiment words and number of negative sentiment words in a note. We then computed the ratio of the number of positive sentiment words to the total number of words in a note. This ratio and the number of positive sentiment words were used as

approach and hand classify the advisor notes to create a training dataset to predict dropout. It would be interesting to compare this type of approach and the lexicon based approach to determine if the expense of hand curating a training dataset is worth it. Further, we could combine the features extracted from the advisor notes with other traditional features such as GPA, SAT scores etc. to improve our prediction accuracy.

6. CONCLUSION

Unstructured data captured in various databases across the educational institution, including in online learning platforms (e.g. Blackboard), are a treasure trove of information that has not been adequately exploited to help the student in improving performance and avoiding dropout. Our study was an attempt at utilizing a small part of this unstructured data to help in the early identification of at-risk students. The fairly high level of prediction accuracy obtained in our study, even without much performance tuning, demonstrates the value of unstructured textual data in institutional databases for detecting at-risk students by predicting student dropout.

Future research should focus on unlocking the potential of unstructured data in institutional databases in helping the student. Other forms of unstructured data such as images, videos, audio clips, illustrations etc. that are created by students for different courses should also be used to extract information that could help provide early intervention and improve student retention.

7. REFERENCES

- [1] Cabrera, A. F., Nora, A., & Castaneda, M. B. (1993). College persistence: Structural equations modeling test of an integrated model of student retention. *The Journal of Higher Education*, 64(2), 123-139.
- [2] Caison, A. L. (2007). Analysis of institutionally specific retention research: A comparison between survey and institutional database methods. *Research in Higher Education*, 48(4), 435-451.
- [3] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321-357.
- [4] Crossley, S., Paquette, L., Dascalu, M., McNamara, D. S., & Baker, R. S. (2016, April). Combining click- stream data with NLP tools to better understand MOOC completion. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge* (pp. 6-14). ACM.
- [5] De Choudhury, M., Gamon, M., Counts, S., & Horvitz, E. (2013). Predicting depression via social media. *ICWSM*, 13, 1-10.
- [6] Delen, D. (2010). A comparative analysis of machine learning techniques for student retention management. *Decision Support Systems*, 49(4), 498-506.
- [7] Drummond, C., & Holte, R. C. (2003, August). C4. 5, class imbalance, and cost sensitivity: why under- sampling beats over-sampling. In *Workshop on Learning from Imbalanced Datasets II* (Vol. 11). Washington DC: Citeseer.
- [8] Herzog, S. (2005). Measuring determinants of student return vs. dropout/stopout vs. transfer: A first-to- second year analysis of new freshmen. *Research in Higher Education*, 46(8), 883-928.
- [9] Hochstein, S. K., & Butler, R. R. (1983). The Effects of the Composition of a Financial Aids Package on Student Retention. *Journal of Student Financial Aid*, 13(1), 21-26.
- [10] Lauría, E. J., Baron, J. D., Devireddy, M., Sundararaju, V., & Jayaprakash, S. M. (2012, April). Mining academic data to improve college student retention: An open source perspective. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (pp. 139-142). ACM.
- [11] Liu, B. (2010). Sentiment Analysis and Subjectivity. *Handbook of natural language processing*, 2, 627- 666.
- [12] McFarland, J., Hussar, B., de Brey, C., Snyder, T., Wang, X., Wilkinson-Flicker, S., ... & Bullock Mann, F. (2017). *The Condition of Education 2017*. NCES 2017-144. National Center for Education Statistics.
- [13] Porter, K. B. (2008). Current trends in student retention: A literature review. *Teaching and Learning in Nursing*, 3(1), 3-5.
- [14] Stampen, J. O., & Cabrera, A. F. (1986). Exploring the Effects of Student Aid on Attrition. *Journal of Student Financial Aid*, 16(2), 28-40.
- [15] Thammasiri, D., Delen, D., Meesad, P., & Kasap, N. (2014). A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition. *Expert Systems with Applications*, 41(2), 321-330.
- [16] Tinto, V. (1975). Dropout from higher education: A theoretical synthesis of recent research. *Review of Educational Research*, 45(1), 89-125.
- [17] Tinto, V. (1993). Building community. *Liberal Education*, 79(4), 16-21.
- [18] Wen, M., Yang, D., & Rose, C. (2014). Sentiment Analysis in MOOC Discussion Forums: What does it tell us? In *Proceedings of the 7th International Conference on Educational Data Mining (EDM2014)*.
- [19] Wetzel, J. N., O'Toole, D., & Peterson, S. (1999). Factors affecting student retention probabilities: A case study. *Journal of Economics and Finance*, 23(1), 45-55. *Journal of Economics and Finance*, 23(1), 45-55.

Data Efficient Educational Assessment via Multi-Dimensional Pairwise Comparisons

Hyunbin Loh^{*1}, Piljae Chae^{*1}, Chanyou Hwang¹

¹Riiid! AI Research

{hb.loh, piljae.chae, cy.hwang}@riiid.co

ABSTRACT

The assessment of students based on their tasks is important in Education, and many advanced methods are applied to the field to solve this problem. Many recent neural network approaches involve heavy modeling of the contents and students. However, it is shown that using pairwise comparisons without the direct usage of instance features can show better assessments in the aspect of consistency, and speed. These ideas have been examined in various perspectives since Thurstone proposed the idea of Comparative Judgement(CJ). Whereas CJ requires direct comparisons of instances to obtain the final fit of the label, we give a generalization by proposing a label prediction model which uses the multi-dimensional features of pairwise comparisons. By reducing the cost in label inference, an Education service can provide visualizations of multi-dimensional skill levels for better meta-cognition of the users. Experimental results on the open dataset *EdNet KT1* show that our method gives higher accuracy even without using the actual responses for the model input.

Keywords

Educational Assessment, Adaptive Comparative Judgement, Deep Learning, Pairwise Comparison

1. INTRODUCTION

In the development of Intelligent Tutoring Systems (ITS), student assessment from their tasks and interactions is a central problem. It is shown in general education scope, that student assessment is highly correlated with the improvement of motivation, engagement, and achievement of the students. Especially in ITS, the decisions of tutoring strategies in many cases rely on algorithmic assessments of student performances. Instances of tutoring decisions include providing educational feedback or adjusting the provided contents to the students in the system. For interactive education systems, real-time computation of assessment is required, and various methods are implemented to settle the

computation time problem.

Methods for student assessment have been studied in various aspects, including well established fields such as Item Response Theory(IRT), Cognitive Diagnosis Model, and Knowledge Tracing. In real-world systems, many assessment methods are based on domain expert knowledge, such as Knowledge Graphs, tagging of contents, and expert designed rule based models (such as the Rasch model in IRT). Recently, data-driven methods with less dependency on domain expert knowledge are also widely applied in ITS. Collaborative Filtering approaches such as Matrix Factorization, or Neural Collaborative Filtering are applied to embed users and items for tasks such as student response prediction, and content recommendation. There are also fully data-driven deep neural networks with no domain expert dependencies that are capable of modeling, prediction, assessment, and recommendation problems in Education such as Deep Knowledge Tracing. These methods not only show high accuracy for the target tasks, but are also easier to apply to new domains since they are domain independent.

However, many existing data-driven methods, including neural network models, require large volumes of training data and also require high costs on inference computations for achieving high performance. Methods based on domain experts can have less complexity, making their operating costs low, but developing such methods often requires a high cost on domain experts. This cost problem can be a barrier on providing e-learning services to people in underdeveloped countries, which also results in digital inequality. [8]

In this paper, we propose two assessment models based on pairwise comparisons to solve the assessment cost problem on data, and inference computation aspect. The key concept of the proposed method is to design a multidimensional generalization of Comparative Judgement(CJ). Instead of using each response data for assessment as in many supervised learning models, we only use pairwise comparisons of user responses. This model can be trained using significantly less label data compared to existing data-driven methods. Also, pairwise comparison data can be gathered within the ITS itself without additional cost. This reduces the cost to gather labeled data, and since the proposed model has less complexity, the computation cost for assessment is also reduced.

We evaluate the proposed methods that use pairwise comparison data by comparing the results with other baseline

Hyunbin Loh, Piljae Chae and Chanyou Hwang "Data Efficient Educational Assessment via Multi-Dimensional Pairwise Comparisons" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 633 - 637

models that directly use response data for predictions.

2. RELATED WORKS

2.1 Student Assessment

Various data-driven student assessment methods have been studied by researchers of ITS, where most are based on three main approaches: knowledge tracing models, collaborative filtering based models, and domain knowledge based models. Knowledge Tracing(KT) is used as a term on various assessment methods to model users by their interactions within a tutoring system [3]. Yudelson, et al. [26] suggested a Bayesian model that estimates the student performance by response correctness data. Piech, et al.[13] applied deep neural networks to KT. Some approaches [1] involve the modeling of contents and students, using pre-trained networks trained for different tasks such as BERT[4], or QuesNet [25]. These approaches directly use the response data of users to estimate the student performance.

Collaborative filtering aims to model users and items to predict potential user-item responses based on user-item interaction data [18]. Using the modeled user and item vectors, one can recommend items to a user that have high predicted labels [17]. Where matrix factorization is widely used due to the simple implementation, neural network models are also suggested to capture more complex features in user-item interactions [7]. The authors of [11] suggested a collaborative filtering based approach to predict the probability of a student answering to a question correctly.

Some methods are based on domain expert knowledge such as Knowledge Graphs, tagging of contents, or expert designed rule based models [5]. Martin, et al.[12] proposed a method to use the Bayesian network that reflects rules designed by domain experts. Item Response Theory(IRT) can be applied by tagging items with their difficulty, or knowledge requirements [10], [20].

2.2 Pairwise Comparison based Models

Supervised learning (regression and classification) is the process of predicting labels of instances using the features of instances. The features of instances can be structured (nominal, ordinal), or unstructured(image, text, sound). Some models also utilize features that are not from the instance themselves by *pre-training* methods. Pre-training is to train a model on an unsupervised auxiliary task and use the trained model to perform the supervised main task [6].

However, there are also models that predict the labels using pairwise comparisons of instances, without using the features from the labels. An instance of a pairwise comparison based assessment method is Comparative Judgement (CJ). The concept was first introduced by Thurstone [22] in the context of Psychological assessment. CJ takes the order comparison data (high or low) of instance pairs to fit a 1-dimensional ordered label of instances. This method is especially effective in domains where there is no standardized assessment method, such as essay marking, image quality assessment [24], [15]. For instance, the authors of [15] performed a major experiment asking professional markers to give comparisons, instead of direct markings. Performing Adaptive Comparative Judgement shows better reliability,

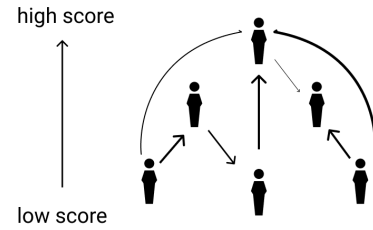


Figure 1: The training step

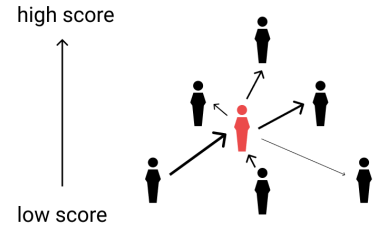


Figure 2: The inference step

and speed than traditional marking in particular areas such as essay marking [16], and mathematics problem solving [9]. Comparative judgement techniques are also applied in areas such as Psychology, Education [19], [9], [14]. The authors of [23] designed a neural network from pairwise comparison data to solve a regression problem, where the features and labels are uncoupled. They make pairwise comparisons of 1-dimensional values to predict the label. We generalize this idea to build models that use multi-dimensional features of pairwise comparisons to predict the label.

3. PROPOSED METHODS

We propose a method to predict the label value of a user. It reflects the responses of the user to items, using pairwise comparisons of responses with other reference users. Possible examples of the labels are preferences on items, expected response values, performance levels on a task, or knowledge levels. The main idea is to model the relative relation of user pairs by the features of pairwise comparisons, as in Figure 1. The arrows between users describe pairwise comparison results, and two users with no arrow in between are incomparable users. For inference, the comparisons of a target user to multiple reference users is used to predict the label of the target user, as in Figure 2.

3.1 Data Description

We use the *EdNet KT1* open dataset [2], which has 95M rows, with columns *userId*, *questionId*, *correctness*, *timestamp*. We do not use any other data source for label data for the experiments to be reproducible from open data. Therefore, note that the following steps to construct labels are not an essential part of the proposed method. If labels of users are available in another experiment setting, then those labels can be used without this additional process. The labels from *EdNet KT1* for this experiment are constructed from the response data by the following procedure:

We sort the items by their total count in the response dataset in decreasing order. Then, we take the first 50 items. Filter the raw data by users who responded to all 50 items, and compute the correctness rate of 50 items and use it as the

label. Filter the raw data by users who responded to all 50 items, and filter out the responses on the chosen 50 items for the experiment.

3.2 Proposed Methods

Before we introduce the details of the proposed models, we describe an example case of the models to illustrate the underlying idea. Consider a case where we have user-item interaction data with columns *userId*, *itemId*, *correctness* as in the *EdNet KT1* data case. Fix two users u_1, u_2 and let TT, TF, FT , and FF be the number of items that both u_1, u_2 responded correctly, only u_1 responded correctly, only u_2 responded correctly, both u_1, u_2 responded incorrectly respectively. If $TT = 90, TF = 10, FT = 110, FF = 40$, then u_1 correctly responded to 90% of the items that u_2 correctly responded, where u_2 correctly responded to 45% of the items that u_1 correctly responded. This relation shows an aspect that the knowledge of u_1 includes the knowledge of u_2 more than the other way round. Then, let y_1, y_2 be some label that reflects the educational performances of users u_1, u_2 . Then, we can consider a model that takes TT, TF, FT, FF and y_2 as features to predict the label y_1 . This model is an example of the first proposed model that we introduce later in this section. The second model that we propose is based on comparisons with multiple users. The main idea is to predict a label from multiple comparisons with other reference users. Now we describe the details of the pre-processing procedure for the proposed models in the general setting.

Consider the general case where we have response data with columns: *user_id*, *item_id*, *response*, where the possible responses of users to items are $1, \dots, r$. We show sample tables for each step of the whole process starting from Table 1.

user_id	item_id	response
1	19	1
1	23	r
2	77	2

Table 1: Raw data example

Fix N items to use as the labels, and filter out responses which have *item_id* in those N items. Group by *user_id* and make r arrays l_1, \dots, l_r where each l_i is the array of *item_ids* that is responded as i . Then, append the label columns y_1, \dots, y_N to this table.

user_id	l_1	...	y_1	...	y_N
1	[19,35,63]	...	0.84	...	0.72
2	[4,19,88]	...	0.30	...	0.54
3	[9, 17]	...	0.76	...	0.66

Table 2: User Table Example

This table has columns *user_id*, $l_1, \dots, l_r, y_1, \dots, y_N$. Now, we fix *reference users*, which is a subset of the users in the User Table. Then, filter the User Table by the users in the *reference users*.

Join the User Table with the Filtered User Table by the *user_id* column of each table to obtain the table with columns *user_id_1*, *user_id_2*, and

$$l_{1,1}, \dots, l_{r,1}, l_{1,2}, \dots, l_{r,2}, y_{1,1}, \dots, y_{N,1}, y_{1,2}, \dots, y_{N,2}.$$

user_id	l_1	...	y_1	...	y_N
1	[19,35,63]	...	0.84	...	0.72
16	[2,64,85]	...	0.89	...	0.78
22	[100,101]	...	0.24	...	0.42

Table 3: Filtered User Table Example

Then, for all $1 \leq i, j \leq r$, append the lengths of the intersections of the array pairs $l_{i,1}, l_{j,2}$ as $x_{i,j}$. Drop the columns *user_id_1*, *user_id_2*, and $l_{1,1}, \dots, l_{r,1}, l_{1,2}, \dots, l_{r,2}$, which finally leaves only the following columns:

$$x_{1,1}, \dots, x_{r,r}, y_{1,1}, \dots, y_{N,1}, y_{1,2}, \dots, y_{N,2}.$$

$x_{1,1}$	$x_{1,2}$...	$x_{r,r}$...	$y_{1,1}$...	$y_{N,2}$
25	42	...	34	...	4	...	12
6	22	...	72	...	10	...	28
15	34	...	2	...	1	...	40

Table 4: Pair Table Example

We call this table with $r^2 + 2N$ columns the Pair Table, and we use this table for model training, where the feature columns are $x_{1,1}, \dots, x_{r,r}, y_{1,1}, \dots, y_{N,1}$, and the label columns are $y_{1,2}, \dots, y_{N,2}$.

Now we introduce the proposed models: PC_1 , and PC_M . The first model PC_1 is a model that predicts the label of a user by comparison with a single other user. The model uses $x_{i,j}$ and $y_{k,1}$ for features, where $i, j = 1, \dots, r$, and $k = 1, \dots, N$. The N -dimensional labels are $y_{k,2}$ for $k = 1, \dots, N$. When $r = 2$, we call $TT = l_{1,1}, TF = l_{1,2}, FT = l_{2,1}, FF = l_{2,2}$. Note that each row of the input data is a comparison with one other user.

The second model PC_M uses pairwise comparisons $l_{i,j,k}$ for M multiple users $k = 1, \dots, M$ as features. The labels are the columns y_1, \dots, y_N of a fixed user. In this model, each row of the input data is the collection of comparisons with multiple users. Then, the loss function is computed by the L1 norm of the N dimensional prediction error. We used a simple fully connected network structure:

- FC($N + r^2$, 64), ReLU
- FC(64, 32), ReLU
- FC(32, N)

3.3 Inference using the proposed model

To predict the labels of a new user u with responses l_1, l_2, \dots, l_r , we compute the join of this row with the User Table to make the Pair Table of u , by following the steps in the previous subsection. The created Pair Table is the table of pairwise comparisons of user u , with the other users in the User Table. Feeding the processed features to the proposed models PC_1, PC_M give the predictions of the labels.

3.4 Neural Network baseline models

Our baseline model is a simple neural network model based on *Fully connected feed-forward network*. The network parameters are set to match the network we proposed above

in 3.2. The hyperparameters, which include the model dimension and depth, have been fit to yield the best results. The Neural Network baseline model takes all user responses as input. We name this model *NaiveFC*. In *EdNet KT1* dataset, there are 3 possible labels for each question. 1 for correct response, 2 for incorrect response and 0 if there is no response. Each value is embedded into a latent space, and the embedded values are added as an input to the model. We find the best performance when the latent space dimension is 128.

3.5 Matrix Factorization and Random Forests as Baseline

We also train a Matrix Factorization (MF) model using Alternating Least Squares [21]. In the proposed models, we split users into train/validation, or train/validation/test. However, matrix factorization models cannot be trained to optimize the results on the validation set, since the user-item embedding of only the validation set will be trained. This leaves the data on train users to be ignored. Therefore, we train the MF model on *train + validation* users and evaluate on *validation* users. This method of data feeding gives higher accuracy since the validation data is included in the train data. Therefore, Matrix Factorization is not a proper baseline for direct comparison because of train-test data cheating. Still, the results are listed as a comparison of the proposed models. We also train a Random Forest Regression model, with maximum depth 30 and the number of trees 300. Likewise, any other regression model can be used after the pre-processing steps.

4. EXPERIMENTS AND RESULTS

From the *EdNet KT1* dataset, we filter the responses from the users who have solved all the 50 most-responded items. The filtered data consists of 9,539,455 responses from 3692 users. To evaluate the data efficiency of our model, we compare the performance of our model while varying the minimum number of responses of each user in the data. The minimum number is varied by 50, 100, and 200, which is the number of responses after excluding the responses for the 50 items. For each setting, the total dataset is filtered by the users who responded more than the minimum number. Then, the users are split into *train* and *test* users by 9:1.

In the PC_M case, we construct three cases for the size of reference users, which are 8/9, 1/9, and 1/90 randomly sampled users from the *train* users in the filtered dataset. These numbers correspond to 80%, 10%, and 1% of the total users. The PC_M models are named by the portion of response users, and the minimum number of responses excluding the label items. For instance, PC_M50 80% corresponds to the case where the users are filtered by those who answered more than 50 questions excluding the label items, and 80% of the total users are randomly assigned as reference users.

We compare the results with the baseline models *NaiveFC*, Random Forest, and a constant value model. Both *NaiveFC* and Random Forest models take the vector with each element representing the response value of non-label items as input. There are 10782 columns that are used as features, since there are 10832 items in total. The constant value model predicts everything as the average of the labels of the

training dataset. The models are trained to predict the label column, which is the correctness rate for 50 label items.

All proposed models based on pairwise comparisons show better performance compared to the baseline models *NaiveFC*, Random Forest, and Matrix Factorization. From the experiments on PC_M , we show that the reduction of 98.75% reference users, also resulting in 98.75% reduction of the feature columns in a different sense, shows similar levels of performance. The 1% reference user case, where there are only 32 reference users, surpasses the performance level of the baseline models, and also shows similar level of performance with more reference users.

The first Matrix Factorization model is trained by the test data included in training, and evaluated by test data. The second model is trained by test data, and evaluated by the same test data. When predicting all the values as the average label values of the training dataset, the MAE for test dataset is 0.1498.

Model	MAE _{train}	MAE _{test}
PC_1100	0.0956	0.0974
PC_1200	0.0955	0.0974
PC_M50 80%	0.0956	0.0973
PC_M50 10%	0.0956	0.0969
PC_M200 10%	0.0956	0.0974
PC_M50 1%	0.0958	0.0966
PC_M200 1%	0.0954	0.0968
NaiveFC	0.1560	0.1648
Random Forest	0.0379	0.1011
MF(Test in Train)	0.3391	0.2437
MF(Trained by Test)	0.1872	0.1872
Average	0.1519	0.1498

Table 5: EdNet Results

5. CONCLUSION AND FUTURE WORK

We have presented two assessment models PC_1 and PC_M based on pairwise comparisons. Experiments show that the proposed models give good results in the *EdNet KT1* case. The features TT, TF, FT, FF capture the relative ordering of the educational performance, as described in the beginning of Section 3.2.

Our method can be applied in any domain where multi-dimensional features capture a uniform ordering of labels, as in the education assessment case. To apply the methods to other problems, one can simply exploit the pre-processing method described in the paper for different labels. The experiments of this paper use labels constructed from the response data, but note that this process is made before applying the proposed models. By using external labels, one can skip the label constructing process and simply feed the pairwise comparison features to the proposed models.

Also, this paper only presents the performance of our models using user response data of *EdNet KT1* as features. We presented a baseline of our approach. Further experiments can be made on other open data such as *ASSISTment*. In our expectation, by leveraging richer data into the features, such as time spent for solving a problem and user behaviors

during solving problems, the accuracy of the models would be improved. Also, adjoining pairwise comparison features to existing real-world models can be a way to reduce the inference cost, as well as label data gathering cost. We leave this as our future work.

6. REFERENCES

- [1] Y. Choi, Y. Lee, J. Cho, J. Baek, D. Shin, S. Lee, Y. Cha, B. Kim, and J. Heo. Assessment modeling: Fundamental pre-training tasks for interactive educational systems, 2020.
- [2] Y. Choi, Y. Lee, D. Shin, J. Cho, S. Park, S. Lee, J. Baek, B. Kim, and Y. Jang. Ednet: A large-scale hierarchical dataset in education, 2019.
- [3] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.
- [4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [5] J.-P. Doignon and J.-C. Falgagne. Spaces for the assessment of knowledge. *International journal of man-machine studies*, 23(2):175–196, 1985.
- [6] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.
- [7] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.
- [8] J. Hvorecký. Can e-learning break the digital divide? *European Journal of Open, Distance and E-Learning*, 7(2), 2004.
- [9] I. Jones, M. Swan, and A. Pollitt. Assessing mathematical problem solving using comparative judgement. *International Journal of Science and Mathematics Education*, 13(1):151–177, 2015.
- [10] B. W. Junker and K. Sijtsma. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 25(3):258–272, 2001.
- [11] K. Lee, J. Chung, Y. Cha, and C. Suh. Machine learning approaches for learning analytics: Collaborative filtering or regression with experts? In *NIPS Workshop, Dec*, pages 1–11, 2016.
- [12] J. Martin and K. VanLehn. Student assessment using bayesian nets. *International Journal of Human-Computer Studies*, 42(6):575–591, 1995.
- [13] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in neural information processing systems*, pages 505–513, 2015.
- [14] A. Pollitt. Let’s stop marking exams. In *IAEA Conference, Philadelphia*, 2004.
- [15] A. Pollitt. The method of adaptive comparative judgement. *Assessment in Education: principles, policy & practice*, 19(3):281–300, 2012.
- [16] A. Pollitt and C. Whitehouse. Using adaptive comparative judgement to obtain a highly reliable rank order in summative assessment. 2012.
- [17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.
- [18] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.
- [19] N. Seery, D. Canty, and P. Phelan. The validity and value of peer assessment using adaptive comparative judgement in design driven practical education. *International Journal of Technology and Design Education*, 22(2):205–226, 2012.
- [20] W. F. Strout. A new item response theory modeling approach with applications to unidimensionality assessment and ability estimation. *Psychometrika*, 55(2):293–325, 1990.
- [21] Y. Takane, F. W. Young, and J. De Leeuw. Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. *Psychometrika*, 42(1):7–67, 1977.
- [22] L. L. Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.
- [23] L. Xu, J. Honda, G. Niu, and M. Sugiyama. Uncoupled regression from pairwise comparison data. In *Advances in Neural Information Processing Systems*, pages 3994–4004, 2019.
- [24] L. Xu, J. Li, W. Lin, Y. Zhang, Y. Zhang, and Y. Yan. Pairwise comparison and rank learning for image quality assessment. *Displays*, 44:21–26, 2016.
- [25] Y. Yin, Q. Liu, Z. Huang, E. Chen, W. Tong, S. Wang, and Y. Su. Quesnet: A unified representation for heterogeneous test questions. *arXiv preprint arXiv:1905.10949*, 2019.
- [26] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *International conference on artificial intelligence in education*, pages 171–180. Springer, 2013.

Using Edit Distance Trails to Analyze Path Solutions of Parsons Puzzles

Salil Maharjan
Ramapo College of New Jersey
smaharj3@ramapo.edu

Amruth N. Kumar
Ramapo College of New Jersey
amruth@ramapo.edu

ABSTRACT

We propose edit distance trail as a representation for analyzing the behavior of students solving Parsons puzzles. The edit distance of a student's solution from the correct solution of a Parsons puzzle gives the degree of correctness of the student's solution. Edit distance trail of a student's solution is the chronological sequence of edit distances of the student's solution from correct solution after each puzzle-solving action. We used edit distance trail representation to analyze the puzzle-solving behavior of students who used a Parsons puzzle tutor on `while` loops. In order to find patterns in student solutions, we applied k-means clustering with elbow method. We found that the centroid curves of the clusters of complete solutions differed by slope, corresponding to the degree of optimality of student solutions. Students found the final few steps of the solution to be more challenging. Centroid curves of clusters of incomplete solutions separated informed attempts from uninformed attempts and identified when students hit a dead-end. We discuss the advantages and drawbacks of our representation as compared to aggregate graphs used in literature and how edit distance trails can provide insight not afforded by descriptive statistics.

Keywords

Edit-Distance, K-means clustering, Patterns in puzzle solutions

1. INTRODUCTION

In a Parsons puzzle [4], the student is given a program with its lines scrambled and asked to reassemble the lines in their correct order. The student is also asked to delete one or more distracters – lines of code that do not belong in the program. These puzzles are rapidly gaining popularity in introductory programming courses. Students preferred solving Parsons puzzles to answering multiple choice questions or writing code in electronic books [3]. Educators like Parsons puzzles because solving puzzles takes significantly less time than debugging code or writing equivalent code, but in one study, it resulted in the same learning performance and retention [2]. Scores on the puzzles were found to correlate with scores on code-writing exercises in another study [8]. Software to administer Parsons puzzles have been developed for programming languages such as Turbo Pascal [4], Python

(e.g., [7,10]) and C++/Java/C# [1].

The sequence of actions taken by students to solve Parsons puzzles can potentially yield insight into their puzzle-solving strategies, just as similar analysis has been proposed for code-writing tasks (e.g., [5]). If patterns can be found in how students go about solving the puzzles, the patterns may in turn be used to predict the likelihood that a student can successfully solve a puzzle, and to provide customized feedback that helps a struggling student get back on track to correctly solve a puzzle. In other words, analyzing how students solve these puzzles could be beneficial to educators, students and researchers.

Whereas the solution to a Parsons puzzle is a state, the sequence of actions taken to solve a puzzle is a path. To date, to the best of our knowledge, only one study has been carried out to analyze the path taken by students to solve Parsons puzzles [6]. In the study, researchers built a visualization of the solution paths used by students and found wide variance among student solutions. They built aggregate graphs of all the solution paths for each puzzle. In the graphs, nodes were puzzle states, the size of each node being proportional to the number of students who had visited that state. The nodes were color-coded based on correctness. Similarly, edges represented state transitions in student solutions, with the width of each edge proportional to the number of student solutions that included the transition. The researchers found sub-optimal puzzle-solving behaviors such as backtracking and circular looping that could be targeted with customized feedback.

This analysis based on puzzle-states can yield puzzle-specific patterns, such as the statement(s) in a puzzle that students have the most trouble assembling. But, since a puzzle with n statements can have $n!$ states, the aggregate graph of the puzzle can be sparse, making it harder to find patterns in student solutions. For the same reason, this approach does not scale well to larger puzzles, i.e., puzzles with more lines of code.

2. EDIT DISTANCE TRAILS

As an alternative to aggregate graphs, we propose to use **edit distance trail**. An edit distance trail is the sequence of edit distances of student's solution from correct solution, one edit distance per action taken by the student to solve the puzzle. In other words, it is a record of edit distances of the student's partial solution from the correct solution from start to finish. In order to find patterns in student solutions of a puzzle, we propose to use **k-means clustering** of edit distance trails with elbow method for determining the value of k .

The operations allowed in a Parsons puzzle are 1) insertion of a statement into the solution 2) deletion of a statement from the solution and 3) reordering of a statement within the solution. The edit distance of a student's solution from the correct solution is

Salil Maharjan and Amruth Kumar "Using Edit Distance Trails to Analyze Path Solutions of Parsons Puzzles" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 638 - 642

the number of these actions necessary to reach the correct solution from the student solution.

In order to calculate edit distances, we modified Levenshtein's algorithm [11]. Levenshtein's algorithm calculates edit distance based on three operations: insertion, deletion and substitution. Since substitution is not an operation permitted in Parsons puzzles, but reordering is, we modified the algorithm to eliminate substitution and incorporate reordering operation.

Edit distance trail of a student on a puzzle is the chronological list of edit distances of the student's partial solution from the correct solution after each action taken by the student to solve the puzzle.

- The starting edit distance of an empty student solution from the correct solution is equal to the number of lines in the puzzle. So, every edit distance trail starts with a value equal to the number of lines in the puzzle.
- When the student's solution is complete and correct, its edit distance from the correct solution is 0. So, every edit distance trail of a completed solution ends with 0.
- The length of the trail is one more than the number of actions taken by the student to solve the puzzle, the extra element corresponding to the start edit distance before the student has taken the first action to solve the puzzle.
- Since our modified Levenshtein's algorithm to compute edit distance treats insertion, deletion and reordering as single-cost operations, each insertion and deletion action increases or decreases the edit distance by exactly 1. A reordering action may change the edit distance by 0 if reordering is incorrect, or 1 if correct.
- If the student inserts a distracter into the solution, edit distance increases by 1.

Unlike the combinatorially explosive number of states in aggregate graphs [6], the length of edit distance trail is linear in the number of actions taken by the student to solve the puzzle. The result of this smaller state space is greater overlap among student solutions, making patterns in student solutions easier to find. Since edit distances abstract away puzzle-specific details such as program states and individual lines in a puzzle, edit distance trails are also amenable to comparison across puzzles.

3. A STUDY OF EDIT DISTANCE TRAILS

We used edit distance trails to analyze the data generated by a suite of tutors on Parsons puzzles called epplets (epplets.org) [1]. The tutors are adaptive and use pretest-practice-post-test protocol – every student solved all the pretest puzzles, but the tutors adaptively selected practice and post-test puzzles based on the learning needs of the student. Students used a drag-and-drop interface to solve puzzles.

For this study, we analyzed the data collected by an epplet on `while` loops. In the epplet, during pretest, students solved Parsons puzzles on the following concepts:

1. A puzzle containing a single `while` loop. The problem (id 2005) on which the puzzle was based was: "A program that reads numbers till the same number appears back to back. It prints the first number to appear twice back to back (e.g., 4 appears back to back in 3,7,5,7,4,4,5 and is printed)."
2. A puzzle containing nested `while` loops, the inside `while` loop's condition dependent on the execution of the outside

`while` loop. The problem (id 2105) on which the puzzle was based was: "A program that repeatedly reads a positive number, reads additional numbers till its multiple is found, and prints the number and its multiple. It repeats this until 0 or negative value is entered for the number. For example, while reading the sequence 3,2,4,6,2,5,4,0 it prints 3,6 and 2,4."

The tutor was used by introductory programming students as after-class assignments. For this study, we used the data collected by the tutor over eight semesters: Spring 2016 – Fall 2019. We included data from only the students who gave permission for their data to be used for research purposes. Students used the tutor in four different languages: C, C++, Java and C#. We combined the data from all four languages in our analysis. Students could use the tutor as often as they wished. When a student used the tutor multiple times, data from all the sessions was included in the study. In all, 1068 students used the tutor during those eight semesters.

Epplets log the sequence of puzzle-solving actions taken by each student. We processed these logs to reconstruct the partial solution after each action and compute the edit distance of the partial solution from the correct solution using modified Levenshtein's algorithm. After computing the edit distance trail corresponding to each sequence of puzzle-solving actions, we used k-means clustering to find patterns in the edit distance trails of the two puzzles (ids 2005 and 2105) separately. Within each puzzle, we analyzed edit distance trails of complete and incomplete solutions separately. The number of edit distance trails available for each puzzle and the optimal number of clusters found for each puzzle for complete and incomplete solutions are listed in Table 1.

Table 1. Number of Edit Distance Trails Available and Optimal Number of Clusters Found for each Puzzle

Puzzle No. (Id)	Complete Solutions		Incomplete Solutions	
	Trails	Clusters	Trails	Clusters
1 (2005)	532	3	239	4
2 (2105)	180	3	153	4

3.1 Puzzle 2005

The clusters found for complete solutions of puzzle 2005 are shown in Figure 1, along with their centroids, which are themselves trails. Table 2 lists the three clusters, number of solutions in each cluster, and the minimum, maximum and mean number of actions taken in those clusters to solve the puzzle.

The puzzle contained 13 lines of code and 2 distracter lines. So, all the centroid curves in Figure 1 start at 13. Data points in the figure at 14 or 15 correspond to the start of trails in which students inserted one or both distracters into the solution before inserting any lines of code that actually belonged in the solution. In the figure, each data point is part of one or more trails – when a data point is shared among trails of different clusters, the colors of the different clusters have blended.

Since a puzzle with n lines can be optimally solved with n actions, cluster 1 (leftmost centroid curve in Figure 1) with a mean of 17.20 actions included all the optimal solutions. The centroid curves of the other two clusters have shallower slopes, corresponding to the use of more actions than necessary to solve the puzzles.

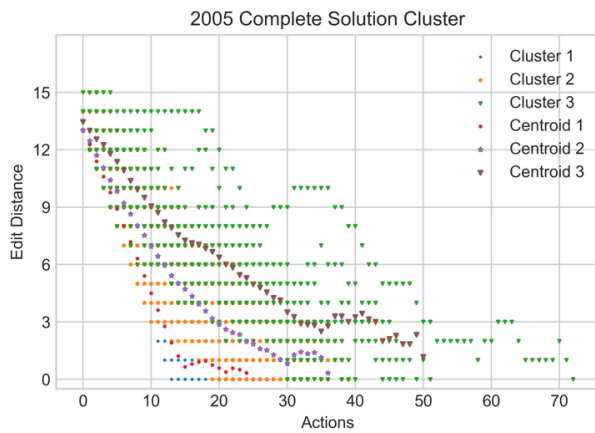


Figure 1. Clusters of Complete Solutions of Puzzle 2005.

Table 2. Complete Solution Clusters of Puzzle 2005 (13 lines):
Number of trails, minimum, maximum and mean actions taken to solve the puzzle

Cluster Number	N	Actions to Solve the Puzzle		
		Minimum	Maximum	Mean
1	383	16	28	17.20
2	112	20	40	27.25
3	24	31	73	40.12

The clusters found for incomplete solutions of the first puzzle are shown in Figure 2. Table 3 lists the number of incomplete solutions in each of the four clusters, the minimum, maximum and mean number of actions taken in the solutions of the clusters and the mean of the final edit distance of all the solutions in the cluster. The final edit distance shows how many more actions were necessary to complete the solution.

Cluster 2 corresponds to student who bailed out after a maximum of 3 actions. It is likely that these students were familiarizing themselves with the user interface of the puzzle and planned to return to use it in seriousness later. Cluster 3 (leftmost centroid curve) comprised of students who made quick progress (mean of 9.90 actions), but reached a plateau at the end before bailing out. They had an average of 9.58 steps left to complete the puzzle. Cluster 1 (second centroid curve from the left) comprised of students who made gradual progress towards the solution (mean of 21.05 actions) before bailing out. The students in this cluster took more actions to solve the puzzle than students in cluster 3, but got closer to the complete solution. Cluster 4 (rightmost centroid curve) was comprised of students who were lost from the beginning. Note that the *slopes of the centroid curves of incomplete solution clusters provide qualitative information about incomplete solution attempts in the cluster*: attempts that were informed (steep slope) versus those that were uninformed and included a lot of redundant actions (shallow slope), and the point at which attempts in a cluster hit a dead-end (plateau).

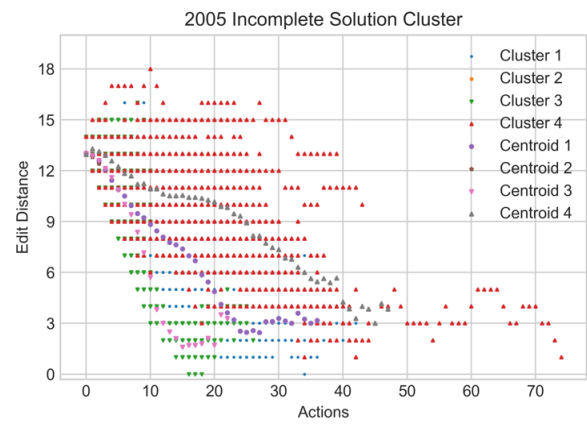


Figure 2. Clusters of Incomplete Solutions of Puzzle 2005.

Table 3. Incomplete Solution Clusters of Puzzle 2005 (13 lines): Number of trails, minimum, maximum and mean actions taken to solve the puzzle

Cluster Number	N	Actions to Solve the Puzzle			Mean final distance
		Min	Max	Mean	
1	78	12	43	21.05	7.05
2	65	1	3	1.50	12.95
3	53	4	27	9.90	9.58
4	38	23	75	32.18	8.63

3.2 Puzzle 2105

Figure 3 and Table 4 show the clusters found among complete solutions of Puzzle 2105, which contained 16 lines of code and 2 lines of distracters. So, complete solution edit distance trails started with a value in the range 16-18 and ended with 0.

The leftmost centroid curve corresponds to cluster 1, which contains all the optimal solutions, and yet, has an average of 22.2 actions. The rightmost centroid curve corresponds to cluster 2, wherein, students were able to solve the puzzle but took almost twice as many actions as cluster 1. The middle centroid curve corresponds to cluster 3, which included students between the other two clusters in terms of the mean number of actions taken. *Clustering of complete solutions resulted in centroid curves with varying slopes, corresponding to solutions at different levels of optimality.*

Table 4. Complete Solution Clusters of Puzzle 2105 (16 lines):
Number of trails, minimum, maximum and mean actions taken to solve the puzzle

Cluster Number	N	Actions to Solve the Puzzle		
		Minimum	Maximum	Mean
1	112	20	33	22.22
2	36	33	66	44.58
3	42	26	42	33.71

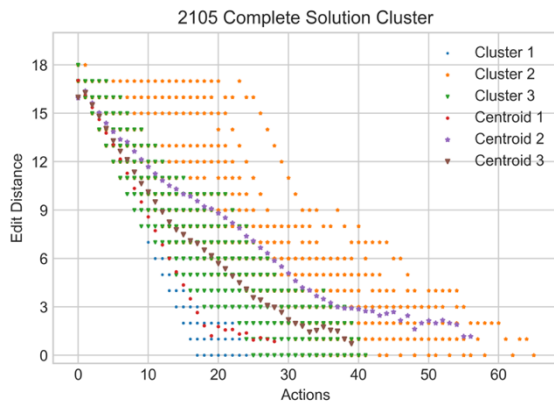


Figure 3. Clusters of Complete Solutions of Puzzle 2105.

Figure 4 and Table 5 show the clusters found among incomplete solutions of puzzle 2105. Cluster 1 represents students who bailed out early, with a maximum of 4 actions. Students in cluster 4 (middle centroid curve) got closest to the correct solution while taking more than twice the number of actions needed to optimally solve the puzzle. Students in cluster 2 (rightmost centroid curve) were less-prepared than those in cluster 4: they took nearly 50% more actions, but were still not as close to the final solution when they bailed out.

Table 5. Incomplete Solution Clusters of Puzzle 2105 (16 lines): Number of trails, minimum, maximum and mean actions taken to solve the puzzle

Cluster Number	N	Actions to Solve the Puzzle			Mean final distance
		Min	Max	Mean	
1	54	1	4	1.24	17.05
2	18	34	84	48.50	10.11
3	26	5	27	10.30	13.34
4	53	18	51	33.88	7.07



Figure 4. Clusters of Incomplete Solutions of Puzzle 2105.

4. DISCUSSION

Analysis of complete solutions of both the puzzles yielded three clusters corresponding to different levels of optimality of the

solution: one cluster corresponding to optimal solutions, and the other two differing in the number of unnecessary actions taken by students to solve the puzzle. Analysis of incomplete solutions yielded four clusters, one of them corresponding to “lurkers” – students who just tried a few actions before bailing out. Lurkers are similar to “stoppers” identified in literature [9] – students who do not take any actions once they encounter a problem, although we believe lurkers were probably just testing the interface. “Movers” identified in literature [9] were all the students in complete solution clusters who were able to solve the puzzle by gradually taking steps towards the correct solution. “Tinkerers” [9] were students in incomplete solution clusters who tried to solve the problem by making small changes in the hopes of making it work.

Edit distance trails can be used to further analyze the behavior of movers and tinkerers. For instance, all the centroid curves in complete solution clusters show a tail at the end of a steep slope. This suggests that even movers who make steady progress solving a puzzle find the last few steps to be more challenging. One possible explanation is that at the end of solving a Parsons puzzle, the student is left with only one or two lines to insert, but the number of locations where they can be inserted are the most ever, making the final steps more challenging.

This illustrates an example of when edit distance trails are more informative than descriptive statistics such as the number of steps taken to solve a puzzle: even in a monotonically decreasing edit distance trail, a change in slope may hint at a moment of frustration (transition from slope to plateau) or insight (transition from plateau to slope). A non-monotonic trail with frequent up-and-down-swings may suggest the use of trial-and-error approach. Such differences may be found in the trails of two different students even when they may have taken the same number of steps to solve a puzzle.

Edit distance trail representation is at a more abstract level than aggregate graph representation reported in literature [6]: using edit distances eliminates puzzle-specific details such as the specific line of code acted upon at each instant by a student. So, aggregate graph representation is better at unearthing puzzle-specific patterns such as determining the specific lines of code that most students might have problems assembling correctly. Edit distance trail representation on the other hand makes it easier to identify patterns among solutions – optimal versus sub-optimal complete solutions, lurking behavior, etc. because of its smaller state space. In the future, with the accumulation of additional data, we hope to find more patterns among complete and incomplete solutions that will provide more qualitative information about the types of solutions.

5. ACKNOWLEDGMENTS

Partial support for this work was provided by the National Science Foundation under grant DUE-1432190.

6. REFERENCES

- [1] Amruth N. Kumar. 2018. Epplets: A Tool for Solving Parsons Puzzles. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*.

- ACM, New York, NY, USA, 527-532. DOI: <https://doi.org/10.1145/3159450.3159576>.
- [2] Barbara J. Ericson, Lauren E. Margulieux, and Jochen Rick. 2017. Solving Parsons problems versus fixing and writing code. In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research (Koli Calling '17)*. ACM, New York, NY, USA, 20-29. DOI: <https://doi.org/10.1145/3141880.3141895>.
 - [3] Barbara J. Ericson, Mark J. Guzdial, and Briana B. Morrison. 2015. Analysis of Interactive Features Designed to Enhance Learning in an Ebook. In *Proceedings of the eleventh annual International Conference on International Computing Education Research (ICER '15)*. ACM, New York, NY, USA, 169-178. DOI: <https://doi.org/10.1145/2787622.2787731>.
 - [4] Dale Parsons and Patricia Haden. 2006. Parson's programming puzzles: a fun and effective learning tool for first programming courses. In *Proceedings of the 8th Australasian Conference on Computing Education - Volume 52 (ACE '06)*, Denise Tolhurst and Samuel Mann (Eds.), Vol. 52. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 157-163.
 - [5] Hosseini, Roya & Hellas, Arto & Brusilovsky, Peter. (2014). Exploring Problem Solving Paths in a Java Programming Course.
 - [6] Juha Helminen, Petri Ihanola, Ville Karavirta, and Lauri Malmi. 2012. How do students solve parsons programming problems?: an analysis of interaction traces. In *Proceedings of the ninth annual international conference on International computing education research (ICER '12)*. ACM, New York, NY, USA, 119-126. DOI: <https://doi.org/10.1145/2361276.2361300>
 - [7] Nick Cheng and Brian Harrington. 2017. The Code Mangler: Evaluating Coding Ability Without Writing any Code. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*. ACM, New York, NY, USA, 123-128. DOI: <https://doi.org/10.1145/3017680.3017704>.
 - [8] Paul Denny, Andrew Luxton-Reilly, and Beth Simon. 2008. Evaluating a new exam question: Parsons problems. In *Proceedings of the Fourth International Workshop on Computing Education Research (ICER '08)*. ACM, New York, NY, USA, 113-124. DOI=<http://dx.doi.org/10.1145/1404520.1404532>.
 - [9] Perkins, D. & Hancock, Chris & Hobbs, Renee & Martin, Fay & Simmons, Rebecca. 1986. Conditions of Learning in Novice Programmers. *Journal of Educational Computing Research*. 2. 10.2190/GUJT-JCBJ-Q6QU-Q9PL.
 - [10] Petri Ihanola and Ville Karavirta. 2010. Open source widget for parson's puzzles. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education (ITiCSE '10)*. ACM, New York, NY, USA, 302-302. DOI: <https://doi.org/10.1145/1822090.1822178>
 - [11] V.I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, 10, 8, 707-710.

How Does Student Behaviour Change Approaching Dropout? A Study of Gender and School Year Differences

Jessica McBroom, Irena Koprinska and Kalina Yacef
School of Computer Science, University of Sydney, Sydney, NSW 2006, Australia
{jmc6755, irena.koprinska, kalina.yacef}@sydney.edu.au

ABSTRACT

In the context of online education, one important consideration is ensuring learning is equitable for a diverse range of students. In particular, understanding how factors such as gender and age can affect student behaviour is crucial for adapting courses to better suit the needs of these students. In this paper, using data from an online introductory programming course, we apply hierarchical clustering to identify changes in student behaviour as students approach dropping out from a course. By considering how these behavioural trends differ based on gender and school year level, we then discuss how this information can lead to insights to assist in improving equity and educational outcomes.

Keywords

gender equity, age equity, student dropout, student behaviour, computer programming education, behavioural trends, hierarchical clustering

1. INTRODUCTION

In an educational setting, ensuring that all students receive equitable learning opportunities is a challenge of great significance. This is particularly important in the context of online education, where teachers may be unable to personally monitor all students due to large cohort sizes, and where the increased accessibility of course materials can allow for very diverse ranges of students. It is also particularly important in areas where certain groups are under-represented, since inequitable education may discourage students from these groups from entering the field. Recent work on improving educational equity has often focused on improvements at an organisational level, such as through teacher training [11], frameworks for addressing equity challenges [7] or analyses of funding distributions [2].

A particularly promising avenue for improving educational equity is the analysis of student behaviour. In particular, educational data mining and analysis techniques can

be utilised to understand how students from different backgrounds respond differently to a course, thereby providing insight into how the course can be modified to improve learning outcomes and equity. Previous work employing such techniques has considered, for example, differences in behaviour at school and home [4], course enrolment and completion rates [10], social behaviour [3], online participation and activity [6, 9], debugging techniques used [8] and motivation for study [5] for different student groups.

In contrast to previous work, this paper uses a hierarchical clustering technique to analyse the evolution of behaviour of students who drop out of an introductory programming course. In particular, samples of student behaviour are taken at different stages during a course (e.g. when they first begin, at points midway through and just before they drop out). These samples are then clustered to detect changes over time and to compare students of different gender and grade groups.

2. DATA

Our data come from a beginner-level Python programming course run in 2018. The course was run online for school students primarily in Australia over a 5 week period, and consisted of weekly exercises interleaved with notes on different topics. In total, this amounted to 40 exercises. Of the 6516 students who attempted the first exercises of the course, 82% dropped out before completing the last exercise.

3. METHODOLOGY

To observe how student behaviour changed as students came closer to dropping out, we analysed the behaviour of all students who completed at least 10 exercises in the first four weeks of the course but still ended up dropping out. These students were selected because they were more likely to be seriously attempting the course (10 exercises constituted 25% of the course), so their dropout was particularly significant. In addition, there would have been more opportunities for interventions to assist these students, so insights from their behaviour could potentially have a larger impact on similar students in future. We considered the first four weeks of the course since all the exercises were comparable (i.e. structured similarly with a similar time limit to complete them). In total, 3677 students were selected.

For each of these students, we then selected a sample of evenly spaced out exercises from the set of exercises they completed during this time, which would represent their

Jessica McBroom, Irena Koprinska and Kalina Yacef "How Does Student Behaviour Change Approaching Dropout? A Study of Gender and School Year Differences" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 643 - 647

behaviour at different stages during their interaction with the course (e.g. when they first began, midway through or just before dropping out). Since all of the selected students had completed at least 10 exercises, we used a sample of 10 evenly spaced out exercises for each student, which was the maximum we could select without including missing data for some students. Since the first and last completed exercise are particularly important when analysing dropout, we wanted to include both of these for each student. As such, we selected exercises using the following process: for each student let $e_1, e_2 \dots e_n$ be the n exercises they completed ($n \geq 10$). Then, define $f(k) = \text{round}(1 + \frac{k(n-1)}{9})$ and select $e_{f(0)}, e_{f(1)}, \dots, e_{f(9)}$. For example, if a student completed 20 exercises, the selected exercises would be $e_1, e_3, e_5, e_7, e_9, e_{12}, e_{14}, e_{16}, e_{18}$ and e_{20} .

After selecting 10 representative exercises for each student, we then generated features to describe their behaviour during each of these exercises, as shown in Table 1. These features related to the number of times particular events occurred, such as viewing the exercise or failing it, and the timings of these events. Note that these features did not need to be independent due to the clustering technique used.

Table 1: Features used to perform the clustering

Feature	Description
num views	the number of times the student viewed the exercise page
num autosaves	the number of times the student's work was autosaved (this was triggered if they had unsaved work that was not modified for 10 seconds)
num failed	the number of times the student submitted their work for marking but did not pass all automated tests
earliest: view, autosave, failure and pass	the time of the first view, autosave, failure or pass respectively (in seconds, relative to the deadline)
average time between fails	if the student failed the exercise two or more times, the average time between these failures, in seconds
time from first failure to completion	if the student ever failed the exercise, the time in seconds from this point until these passed.

After preparing the features, we applied the temporal hierarchical clustering algorithm DETECT [1] to find clusters of student behaviour that changed over time. This algorithm produces hierarchical clusters defined by decision rules (e.g. a cluster could be all cases where the number of views was ≤ 3 and the number of fails was > 2). To do this, it performs a search over many different options for clustering the data, each time observing the resulting distribution of clusters over time. It then chooses the option that maximises an objective function based on this distribution. In this case, the objective was to find clusters that changed the most between the student's first two sample exercises and their last two. Since the algorithm selects only the best features in the final clustering, the method is robust to dependencies between features. The resulting clusters are shown in Figure 1 and discussed in more detail in the next section.

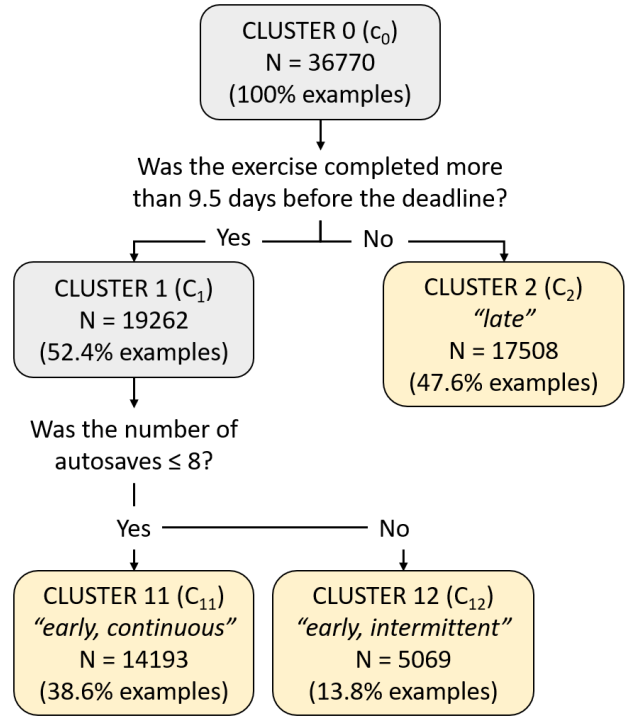


Figure 1: Hierarchical clusters of student behaviour. The cluster names show the hierarchical nature of the clusters. Cluster 2 represents cases where students completed the exercise late. In contrast, Clusters 11 and 12 represent cases where students completed the exercise early and worked continuously and intermittently respectively. This is discussed in further detail in the text.

After producing the clusters, we then analysed the differences in cluster distributions for students from different backgrounds. In particular, we considered the relationship between student gender and school year level on the cluster distributions over time in order to gain insight into equity issues. The results are discussed in the next section.

4. RESULTS AND DISCUSSION

4.1 Behavioural Clusters

The behavioural clusters produced by DETECT are shown in Figure 1. The cluster names indicate the hierarchical nature of the clusters: c_0 at the root contains all examples, c_1/c_2 are mutually exclusive subsets of c_0 and c_{11}/c_{12} are mutually exclusive subsets of c_1 . Since there were 3677 students in total and 10 representative examples for each student, this made $N = 3677 \times 10 = 36770$ clustered examples of behaviour in total, with 38.6%, 13.8% and 47.6% in c_{11} , c_{12} and c_2 respectively.

In this work, we focus on the three final clusters, c_{11} , c_{12} and c_2 , which we label for convenience as “early, continuous”, “early, intermittent” and “late” respectively. We label c_2 as “late” since it represents cases where students completed the exercise close to the deadline (i.e. within 9.5 days of it). In contrast, c_{11} and c_{12} are labelled as “early” since here students completed the exercise more than 9.5 days before

the deadline. In addition, c_{11} is labelled as “continuous” because there were ≤ 8 autosaves, and these were triggered when a student with unsaved work paused for more than 10 seconds. As such, a student with a small number of autosaves did not pause very often and worked continuously. In contrast, we label C_{12} as “intermittent” since there were a large number of pauses.

Since we used a sample of 10 exercises for each student, this meant that students could move from cluster to cluster over time. As such, in order to understand how the cluster distributions changed over time, we plotted the number of students in each cluster for each exercise, as shown in Figure 2. Note that the total number of students in the graph is constant over time (3677 students). Note also that the exercises are relative to the students, not the course. For example, Exercise 1 and Exercise 10 represent the first and last exercise that each student completed before dropping out, not the first and tenth exercise in the course.

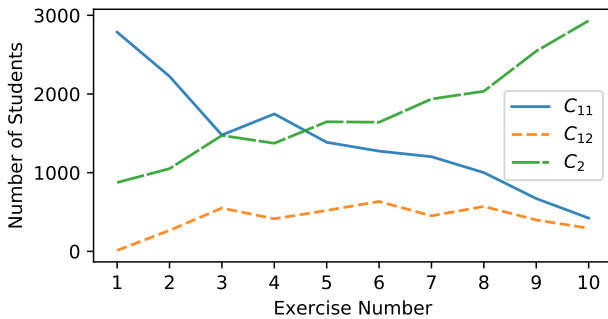


Figure 2: Cluster distributions over time

When submitting their first exercise, most students were in c_{11} (early, continuous), where they completed the exercise more than 9.5 days before the deadline and had ≤ 8 autosaves, indicating that the students worked continuously on the exercise. As such, dropping out students tended to complete the exercises early and continuously when they first began the course. Over time, however, the number of students in the other clusters increased. In particular, by the time they were close to dropping out, most students were in c_2 (late), where they were no longer completing the exercises early. In addition, the increase in c_{12} (early, intermittent) indicates that the students who did complete the exercise early paused more, possibly due to difficulty or distraction.

In summary, by clustering evenly spaced-out samples of student work over time, it is possible to observe how student behaviour develops as students approach dropping out. In the next sections, we will filter these students based on grade level and gender to observe differences in these trends for different student groups.

4.2 School Year Level Differences

In order to analyse the differences between students of different school year levels, we divided students into four groups based on school year, as shown in Table 2. Using the same clusters as before, we then observed the differences in cluster distributions with respect to these groups. The results are shown in Figure 3.

Table 2: Grade groups used in the analysis. N is the number of students in each group who dropped out but completed at least 10 exercises in the first four weeks of the course.

Group	N	Description
Year 11+	263	Senior students in Year 11 or above
Years 9-10	2090	Intermediate students in Year 9 or 10
Years 7-8	1081	Junior students in Year 7 or 8
Primary	238	Primary students in Year 6 or under

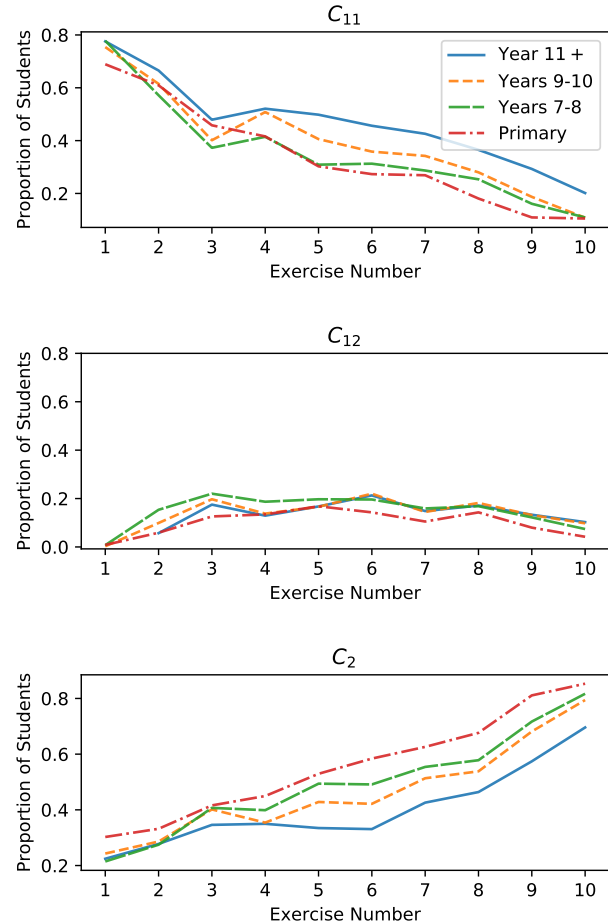


Figure 3: Differences in cluster distributions across school grade groups

From the graphs in Figure 3, one important observation is that increasing age is correlated with a decrease in the proportion of students in c_2 (late), and this difference increases over time. This suggests that younger students tend to complete exercises later than older students as they approach dropping out. This could suggest that younger students are more likely to drop out because they are falling behind and possibly having difficulties with time management, whereas older students may be dropping out for other reasons, such as losing interest or the exercises being too easy.

Another interesting observation is that, while most students were similar at the beginning, older students were more

likely to be in c_{11} over time than younger students. Since c_{11} represents behaviour where students complete the exercise early and work continuously, this suggests that older students may have been more organised than younger students, or found the exercises easier immediately before dropping out. This supports the idea that older students may have been dropping out because the exercises were too easy, while younger students may have done so due to difficulty.

These observations are important for informing future course development, since they can provide insight into how courses can be made more equitable. For example, if younger students are more likely to drop out from a course because it is difficult and they are falling behind, then interventions could be developed to help support these students. For example, they could be given extra practice questions or time to complete the exercises. In contrast, if older students were dropping out because the exercises were too easy, then more advanced content or optional extension exercises could be added for these students. This could then help to make the course more equitable by addressing the needs of different student groups.

4.3 Gender Differences

In addition to analysing differences based on school grade, we also considered how student behaviour differed based on gender. In total, we analysed data from 2334 male students and 1124 female students who dropped out after completing at least 10 exercises from the first four weeks of the course. The differences in cluster distributions over time are shown in Figure 4.

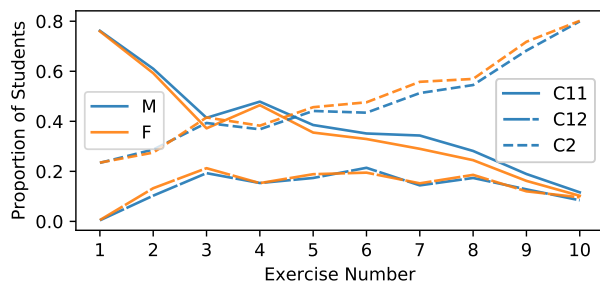


Figure 4: Differences in cluster distributions across gender groups

Interestingly, there is very little difference in the cluster distributions for male and female students over time for this course. This suggests that male and female students behaved similarly with respect to these clusters as they approached dropping out - they started the exercises at similar times and worked roughly as continuously as each other. The dropout rates for male and female students were also similar. Such information is highly valuable for improving gender equity, since it highlights where to focus attention. In particular, instead of comparing the behaviour of male and female students who drop out in order to introduce different types of interventions, perhaps focusing on reducing dropout in general for both groups, or focusing on improving the balance in enrolment rates, could assist in improving gender equity for this course.

5. CONCLUSION

In this paper, we have discussed how the behaviour of students who drop out from a course can be analysed in order to improve equity. In particular, representative samples of work from dropout students can be clustered to identify changes over time. Differences and similarities in trends as students approach dropout can then be observed for different student groups (e.g. male and female students or students from different grade levels). This comparison can lead to insights into the potential reasons for why students dropout, helping to inform further course development to improve equity.

6. REFERENCES

- [1] Anonymous. DETECT: A hierarchical clustering algorithm for behavioural trends in temporal educational data. *Unpublished, submitted to AIED2020. Will be made available for reference after double blind review process.*
- [2] B. Baker and J. Levin. Educational equity, adequacy, and equal opportunity in the commonwealth: An evaluation of pennsylvania's school finance system. Technical report, American Institutes for Research, 2014.
- [3] J. Bayer, H. Bydzovská, J. Géryk, T. Obsivac, and L. Popelinsky. Predicting drop-out from social behaviour of students. In *the 5th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2012.
- [4] G. Ben-Zadok, M. Leiba, and R. Nachmias. Comparison of online learning behaviors in school vs. at home in terms of age and gender based on log file analysis. *Interdisciplinary Journal of E-Learning and Learning Objects*, 6(1):305–322, 2010.
- [5] S. Chopra, H. Gautreau, A. Khan, M. Mirsafian, and L. Golab. Gender differences in undergraduate engineering applicants: A text mining approach. In *the 11th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2018.
- [6] D. A. Fields, Y. B. Kafai, and M. T. Giang. Youth computational participation in the wild: Understanding experience and equity in participating and programming in the online scratch community. *ACM Transactions on Computing Education (TOCE)*, 17(3):1–22, 2017.
- [7] P. Gorski. Rethinking the role of “culture” in educational equity: From cultural competence to equity literacy. *Multicultural Perspectives*, 18(4):221–226, 2016.
- [8] V. Grigoreanu, L. Beckwith, X. Fern, S. Yang, C. Komireddy, V. Narayanan, C. Cook, and M. Burnett. Gender differences in end-user debugging, revisited: What the miners found. In *Visual Languages and Human-Centric Computing (VL/HCC'06)*, pages 19–26. IEEE, 2006.
- [9] J. McBroom, B. Jeffries, I. Koprinska, and K. Yacef. Mining behaviours of students in autograding submission system logs. *the 9th International Conference on Educational Data Mining*, 2016.
- [10] J. McBroom, I. Koprinska, and K. Yacef. Understanding gender differences to improve equity in

- computer programming education. In *Proceedings of the Twenty-Second Australasian Computing Education Conference*, pages 185–194. ACM, 2020.
- [11] M. Wilson. Pre-service teachers’ emerging views on educational equity. Master’s thesis, Eastern Michigan University, 2019.

Measuring task difficulty for online learning environments where multiple attempts are allowed – the Elo rating algorithm approach

Maciej Pankiewicz
Warsaw University of Life Sciences
maciej_pankiewicz@sggw.pl

ABSTRACT

The aim of this research is to examine the accuracy of the estimations performed with the Elo rating system in an online learning environment where multiple attempts are allowed and feedback is provided after every submission. The acquired estimations are compared to the reference difficulty values calculated by the means of the IRT graded response model. The data originates from the RunCode online learning environment (<https://runcodeapp.com>) developed for the purpose of learning programming skills. The platform has been made available to 299 first semester computer science students with varying initial programming knowledge. There have been 50055 attempts on 76 tasks recorded. Multiple attempts on tasks were allowed, there was no penalty imposed for extra tries and feedback was provided after every submission. High correlation values – up to 0.927 – have been observed for the estimations performed by the Elo rating algorithm. We argue that the design of the Elo algorithm makes it a good choice as the on-the-fly task difficulty estimation method for online learning environments where multiple attempts are allowed and feedback is provided after submission.

Keywords

Task difficulty, Elo, rating algorithm, gamification.

1. INTRODUCTION

There are several methods developed for the purpose of estimating task difficulty that originate to a great extent from the area of item response theory. Some aspects make application of these methods in the context of online learning environments difficult, e.g. computational demands or difficult implementation. Therefore alternative methods of the difficulty estimation are analyzed [1], with the focus on lower computational demands and easier implementation. The Elo rating algorithm is an example of such alternative methods that satisfies these requirements, however, often with the cost of lower (reasonably) estimation accuracy. In online learning environments with formative assessment approach – contrary to knowledge assessment systems – lower accuracy of the estimations may be often accepted. Such learning environments may benefit from the implementation of faster

methods, even if the requirement of high estimation accuracy is not met. The Elo rating system has already found several implementations in the educational context [1, 2, 3, 4]. However, most of the up-to-date research focuses on its applications within knowledge testing environments (summative assessment) or online learning platforms (formative assessment) where one attempt is allowed and with examples of task types presenting low complexity, e.g. multi-choice, where it is easy to satisfy the requirement of automated evaluation. The programming assignment is an example of a task type with much higher complexity – it is highly improbable to “guess” the correct answer for such a task type. It is, however, a task type that also satisfies a requirement of an automated evaluation and there are multiple types of automated tests that may be executed on the programming code in order to verify its correctness [5]. It is intuitively expected that on average, the number of attempts needed to correctly solve a programming task is much higher than of e.g. multi-choice task type. But how does it impact the quality of task difficulty estimation? What is the impact of multiple attempts, especially if there is a significant number of tasks available in the system on which learners fail multiple times? Especially online learning environments may benefit from the answer to this question. Dynamically changing number of system users and (or) of collaboratively added tasks make the on-the-fly requirement of the task difficulty estimation hard to satisfy already for a small number of system users and tasks – if using the well-known difficulty estimation methods originating from the area of e.g. item response theory. On the other hand, usage of alternative methods for difficulty estimation may satisfy the on-the-fly requirement, but often with the cost of lower accuracy. This cost however may be often accepted and this research contributes to the question of the above-mentioned compromise between accuracy and on-the-fly calibration requirements in online learning environments.

2. ESTIMATING DIFFICULTY

Models created for the purpose of estimating task difficulty originate mainly from the area of Computerized Adaptive Testing (CAT) domain. These models are used in order to optimize the process of knowledge assessment by lowering the number of tasks and time needed to determine learner's current knowledge level. There are two estimations evaluated: of a task difficulty and of a learner ability. Foundations for the development in this area have been laid by G. Rasch that formulated the single parameter logistic model with difficulty parameter [6, 7, 8]. The model and its variations under the name of the Item Response Theory (IRT) have been since utilized not only in educational [9], but also medical [10] or marketing [11] applications. In the era of the internet education, methods for estimating task difficulty have

Maciej Pankiewicz "Measuring task difficulty for online learning environments where multiple attempts are allowed -- the Elo rating algorithm approach" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 648 - 652

been used not only for the purpose of the knowledge assessment [12], but are increasingly present in the field of Intelligent Tutoring Systems [13] where they are used for the purpose of matching item difficulty to learner ability in order to optimize the process of knowledge acquisition and achieve so called adaptivity. There are several examples of adaptive online learning systems e.g. for learning factual knowledge in the field of geography [1] or mathematics [2]. There are various methods other than item response theory models used for the purpose of task difficulty estimation that have been evaluated within educational research community e.g. Elo rating algorithm [1, 2, 12], proportion correct [12, 14], learner feedback [12, 15] or expert rating [12]. It has been found that the accuracy of these methods may achieve values described as “accurate-enough” for the purpose of online learning environments [1]. Although in terms of requirements for knowledge assessment systems such accuracy may not always be accepted, it may be a reasonable choice for usage within online learning environments. Above mentioned methods present several weaknesses in terms of their usage within online environments, e.g. require large calibration samples or high computational demands (IRT models), require large number of votes (learner feedback) or availability of experts (expert rating), in the context of their usage within online learning environments some of them may be more reasonable than other, depending on specific aspects of an individual system requirements. This article focuses on the usage of the Elo rating system as it is the algorithmic approach and therefore easier to automate than methods that require involvement of a human, e.g. expert rating or learner feedback. Additionally, it has already been validated as a suitable tool for online learning environments [1]. It has been implemented e.g. in the system with multi-choice questions where one attempt is allowed. This research extends the up-to-date research by presenting results of the analysis performed on the example of the online learning environment with the assignment of higher difficulty level (programming assignment), where multiple attempts are allowed, feedback is provided after every submission and no penalty is imposed for extra tries.

3. ELO RATING ALGORITHM

The Elo rating system [16] has been developed for the purpose of measuring strength of players in chess tournaments. The aim of the algorithm is to calculate players’ rating change after every game. That change depends on outcomes of tournament games. Every player is assigned a rating that is usually a number between 1000 and 3000 that is a subject to change after every game. New rating is calculated by a formula:

$$R_n = R + K(O - P)$$

Where: R_n is the new value of the rating, R – the actual rating, O – game outcome (1 – win, 0 – loss), P – probability of winning the game and constant K – the value for chess tournaments is often 32. The probability of winning P is given as:

$$P = \frac{1}{1 + 10^{\frac{R_o - R_p}{400}}}$$

Where R_p is the rating of a player and R_o is the rating of the opponent. In the context of an online learning environment, we consider a tournament game to be a single submission of a

solution, a player – a learner that submits the solution and opponent – a task.

There are three possible outcomes of the chess game (win, loose, draw), but in the context of learning environment we only consider two outcomes: learner wins if the submission receives the maximum score or learner loses if the submission does not receive maximum score.

4. METHODOLOGY

4.1 Programming course

The RunCode online learning environment is an online application that supports automated validation of the correctness of programming code available at <https://runcodeapp.com>. It provides access to various courses consisting of programming assignments that are grouped into modules for the purpose of clarity. There are several gamification enhancements introduced to the platform aimed at keeping the user engaged.

4.2 Programming assignment

Students learn to code using the RunCode online learning environment by solving programming assignments. Every assignment requires a student to create a code containing a function that will be executed by the test runner in order to check its correctness. Task description defines requirements that the function should meet. Students submit the code containing the function and immediately (after its execution by the test runner) receive score and feedback. Score is calculated as the percentage of the tests, that ended with success to the overall number of tests performed on the code and is presented as a value in the range [0%-100%]. The feedback information is based on the information returned by the test runner and contains errors and warnings (if any) returned by the compiler and results of tests executed by the test runner containing information about the correctness of the submitted code. Only submissions with no errors, no warnings and satisfying requirements of all tests defined by the lecturer receive the maximum score (100%). Multiple submissions are allowed and feedback is provided after every submission.

4.3 Data

The data originates from the gamified course available on the RunCode online learning environment: a platform developed for the purpose of learning programming skills. The RunCode system supports automated evaluation of the submitted programming code. The RunCode platform has been made available as an additional, optional tool during the first-semester *Introduction to programming* course at the Faculty of Applied Informatics and Mathematics, Warsaw University of Life Sciences. The course is mandatory for the students of computer science and is realized in a traditional way – with lectures and computer classes. The main online tool for managing the course resources is the university’s moodle website. Although the RunCode platform usage was not mandatory and results obtained were not included in the final grade, majority of the students decided to use the system on regular basis. The course containing 76 programming tasks has been made available on the RunCode application. The data has been collected during two winter semesters: 2017/2018 and 2018/2019. During this period, 299 students with varying initial programming knowledge used the system. There have been 50055 attempts recorded in total. Multiple attempts were allowed with no penalty imposed on extra tries and feedback was provided

immediately after every submission. Students self-elected the order of solving tasks.

5. RESULTS

The data has been collected during two academic years: 2017/2018 and 2018/2019 and contains system usage data that originate from the RunCode platform and results of the survey on the declared initial level of programming knowledge. Before the course started, students took a survey and answered the question about perceived programming skill level – self-evaluation of their knowledge of basic programming concepts. It has been a surprising observation, that about one third of students of the first semester at the Faculty of Informatics declared having completely no previous experience with programming languages and more than a half declared having no (skill level 1) or little (skill level 2) previous programming experience (Table 1).

Table 1. Results of the pre-course survey on programming skill level: 1 – no previous programming experience, 5 – very extensive programming experience.

	1	2	3	4	5
2017 (n=110)	32.7%	24.5%	15.5%	15.5%	11.8%
2018 (n=159)	32.7%	25.8%	22.0%	11.3%	8.2%

The overall engagement of the students, measured as the number of user submissions on the RunCode platform has been presented in Figure 1. The overall engagement of students is considerably high with the average of 178 submissions (attempts) performed by a user.

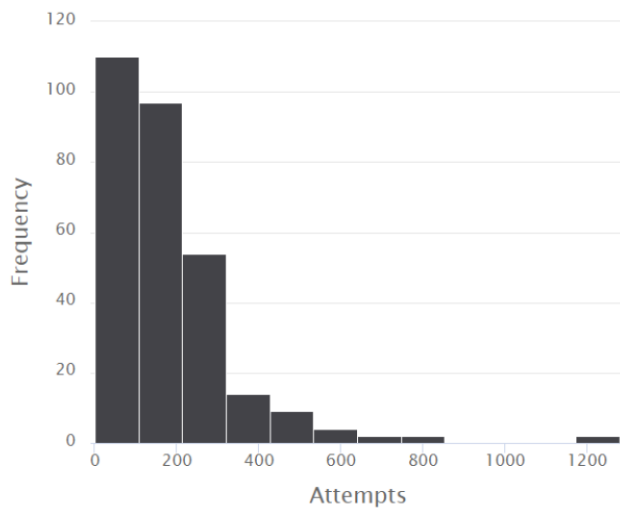


Figure 1. Histogram of the number of attempts performed by users of the RunCode platform.

The following detailed analysis of the submission data (Table 2) for the purpose of clarity has been limited to seven first attempts (ca. 75% of all samples). This limitation is reasonable, as the effects visible on the first seven attempts are in general also reflected in the remaining data (e.g. dropout) with the long tail of even more than 50 attempts on a task. Two important observations may be made basing on the overall view of the data presented in Table 2. Firstly, the number of successful attempts overall is low. On average, the first attempt is correct only in 39% of submissions. If the first attempt was not successful, the success

rate for the second submission is 30% and with following attempts, the success rate decreases.

Table 2. The number of correct and incorrect attempts on assignments. The Total column is the cumulative sum of attempts. The Dropout column is the percentage of students that resigned to take another attempt.

Attempts	Incorrect	Correct	Total	Dropout
1	8259	5269	13528	-
2	5623	2389	21540	0,030
3	4045	1244	26829	0,059
4	2950	842	30621	0,063
5	2259	493	33373	0,067
6	1766	342	35481	0,067
7	1396	237	37114	0,075

It denotes, that the average difficulty level of tasks available on the platform may be perceived as high. Secondly, despite the fact that users fail to upload successful solution on the first attempt, they feel motivated and do not give up. The dropout rate is very low. Only 3% of the system users give up if the first attempt was not successful. As the number of submission increases, the dropout rate increases but even at the 7th attempt is reasonably low (7%). In order to compare the difficulty estimations calculated by the Elo rating algorithm with the reference values the Pearson's correlation coefficient has been used. Reference values for the following analysis have been calculated by the means of the IRT graded response model [17]. The graded response model is suitable for modelling polytomous response data and has been already introduced e.g. for the purpose of knowledge assessment on open-ended tasks with multiple attempts allowed [18]. It has been found that the estimations of the IRT graded response model are accurate already for sample size of $n = 200$ [20]. The encoding procedure of polytomous data for the purpose of this analysis was following: the user-task matrix for the i -th attempt on task n by user m has been filled with value of i , if the first submission was successful. If the second attempt was successful, the value inserted was $i-1$. Every following attempt needed to achieve the maximum score lowered the inserted value by 1. In this scenario if a learner does not succeed in a maximum allowed number of attempts, the inserted value is 0. The procedure does not distinguish between not taking the task and exhausting all available attempts with no success. The study on the effects of missing data on the accuracy of estimations performed by the graded response model may be found e.g. in [19]. The reference (IRT) values for the following analysis have been calculated on the full data set. The optimal value of the Elo uncertainty parameter K has been evaluated experimentally, similarly to [12, 14]. The highest correlation with estimation values calculated with the graded response model has been achieved for the value of $K = 3$. The PlayerRatings R package [21] with default values of the initial rating and rating deviation has been used to perform Elo algorithm calculations and RapidMiner – for the ETL data processing [22]. The highest correlation value – 0.927 – has been observed for cumulative data from three attempts on tasks (Table 3). The correlation calculated only on the data from the first attempt achieves low correlation of the value 0.565.

Table 3. Correlation of the difficulty estimations calculated by Elo algorithm compared to the reference values.

Att.	1	2	3	4	5	6	7
Cor.	0,565	0,887	0,927	0,908	0,892	0,873	0,852

With increasing number of attempts, the correlation decreases – correlation value calculated for the cumulative data from 7 attempts is 0.852 (Figure 2).

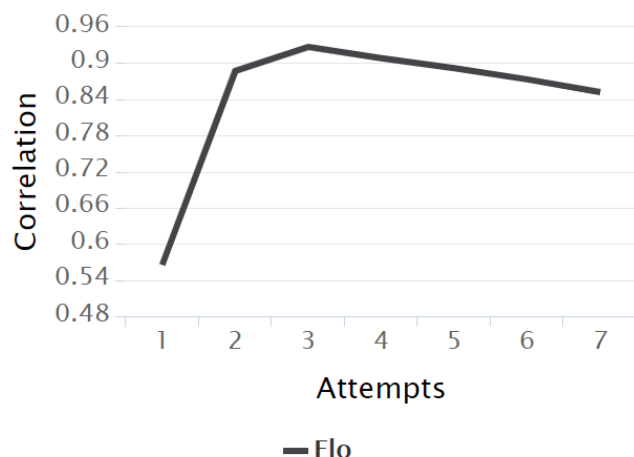


Figure 2. Correlation of the difficulty estimations calculated by Elo algorithm with the reference values.

6. SUMMARY AND DISCUSSION

The aim of this research has been to evaluate the accuracy of the Elo rating algorithm in terms of the task difficulty estimation. The analysis has been performed in order to verify the appropriateness of the method for its usage within online learning environments where multiple attempts are allowed and feedback is provided after every attempt. The source of the data has been the programming course available at <https://runcodeapp.com> – an online application developed for the purpose of learning programming skills. The analysis has been performed on the sample of 50055 attempts on 76 tasks submitted by 299 learners – first semester students of computer science. The data has been gathered during two academic years: 2017/2018 and 2018/2019. Although usage of the platform was not mandatory, a high level of engagement has been observed – the dropout rate for consecutive attempts was in the range of 3-7%. Students presented varying levels of initial programming knowledge, with about one third declaring no previous experience with programming. The highest correlation of 0.927 has been calculated for data containing three attempts on task. With an increasing number of attempts, the correlation value has slowly decreased. The obtained correlation level may satisfy the requirements of the online learning environment and estimations may be perceived as sufficient. Similar values of correlation have been already obtained in previous research – does it mean that the Elo rating algorithm may be a reasonable choice for estimating difficulty within online learning environments? Under circumstances described in the following, it may be. Contrary to online assessment applications where large calibration samples are required, requirements of online learning environments in terms of the accuracy may not be that strict – although delivering lower accuracy, the Elo algorithm is quick and it is the main advantage. Novel aspects of this analysis concern following factors: 1) it is based on the data

originating from the real online learning environment created for the purpose of fostering basic programming skills; 2) allowance of the multiple (unlimited) attempts on task and feedback provided after every submission; 3) high level of the task difficulty observed as the large average number of attempts required to complete the task. There are several considerations that may limit the interpretability of the results and their generalization that may be divided into three elements referring to the RunCode platform, users and task characteristics. The RunCode online learning environment is a gamified internet application. The aim of the implemented gamification elements is to engage platform users and motivate them towards reaching the maximum score on every task. Overall engagement of system users may be described as very high and the gamification may be an important source of the large user contribution. The number of students that give up after an unsuccessful attempt is very low and varies between 3% and 7% for the first 7 attempts (Table 2). It should be a subject for further analysis, if the results may be repeated if the number of dropouts in the data increases. The platform has been made available to the first semester students of computer science enrolled in the *Introduction to programming* course. The variety of the skill level is broad in the analyzed group. Although the first impression may be that students enrolled in the computer science track already have experience with basics of programming, student responses in the survey completed at the beginning of the course do not confirm this suspicion. One-third of the students declares to have absolutely no previous experience with any programming language, but there are also several students that have already mastered the basic programming concepts before joining the course. It is to be analyzed, if the observations from this study are repeated if users present equal (e.g. very low) initial knowledge on the subject. It is also to be considered, that the motivation of the computer science student to succeed in the *Introduction to programming* course may be reasonably higher than of an average user that joins any programming course at any publicly available online learning platform. Although usage of the platform was not mandatory and results obtained were not impacting the final grade, students used the platform very extensively. Therefore, it is to be analyzed if the observations made within a group that focused on the success in the course apply in other contexts. The overall difficulty level of the programming assignment available on the RunCode platform may be described as high. The submission process is very complex in comparison to e.g. multi-choice questions. Even easiest tasks (as perceived by the lecturer) received on average a higher number of attempts than initially expected. It may result from the fact that unexperienced learners that joined the course struggled from the beginning with too many new concepts: not only related to the basic rules of code preparation, but also e.g. to the technical aspects of creating code with usage of the integrated development environment (IDE). There is an additional outcome of the large average number of submissions on a task. The difficulty level may be estimated with higher granularity, even if the number of system users is low. On the other hand, if these tasks were made available outside of the university's course, on an online platform to the public, a high average difficulty level would possibly lead to learners' frustration and it would be expected that the dropout rate will be much higher. Future work will be aimed at comparing other methods of difficulty estimation satisfying the requirements of the on-the-fly calibration.

7. REFERENCES

- [1] Pelánek, R., Papoušek, J., Řihák, J., Stanislav, V., & Nižnan, J. (2017). Elo-based learner modeling for the adaptive practice of facts. *User Modeling and User-Adapted Interaction*, 27(1), 89-118. <https://doi.org/10.1007/s11257-016-9185-7>
- [2] Klinkenberg, S., Straatemeier, M., & van der Maas, H. L. (2011). Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, 57(2), 1813-1824. <https://doi.org/10.1016/j.compedu.2011.02.003>
- [3] Papousek, J., Pelánek, R., & Stanislav, V. (2014). Adaptive practice of facts in domains with varied prior knowledge. In *Proc. of educational data mining* (pp. 6-13).
- [4] Pankiewicz, M. & Bator, M. (2019). Elo Rating Algorithm for the Purpose of Measuring Task Difficulty in Online Learning Environments. *e-mentor*, 5(82), 43-51.
- [5] Ala-Mutka, K. M. (2005). A survey of automated assessment approaches for programming assignments. *Computer science education*, 15(2), 83-102.
- [6] Rasch G., 1960, Probabilistic Models for some Intelligence and Attainment Tests, Danish Institute for Education Research, Copenhagen.
- [7] Rasch G., 1966, An individualistic approach to item analysis, [w:] P.F. Lazarsfeld, N.W. Henry (eds.) *Readings in mathematical social sciences*, Cambridge: MIT Press, 89-107.
- [8] Rasch G., 1977, On specific objectivity: An attempt at formalising the request for generality and validity of scientific statements, *Danish Yearbook of Philosophy*, vol. 14, s. 58–94.
- [9] Scheerens J., 2003, *Educational Evaluation, Assessment, and Monitoring: A Systemic Approach*, Swets & Zeitlinger, Lisse–Exton.
- [10] Christensen K.B., Kreiner S., Mesbah M., 2013, *Rasch Models in Health*, ISTE–Wiley, London–Hoboken.
- [11] Bechtel G.G., 1985, Generalizing the Rasch model for consumer rating scales, *Marketing Science*, vol. 4, no. 1, s. 62–73.
- [12] Wauters, K., Desmet, P., & Van Den Noortgate, W. (2012). Item difficulty estimation: An auspicious collaboration between data and judgment. *Computers & Education*, 58(4), 1183-1193. <https://doi.org/10.1016/j.compedu.2011.11.020>
- [13] Wauters, K., Desmet, P., & Van Den Noortgate, W. (2010). Adaptive item-based learning environments based on the item response theory: possibilities and challenges. *Journal of Computer Assisted Learning*, 26(6), 549-562. <https://doi.org/10.1111/j.1365-2729.2010.00368.x>
- [14] Antal, M. (2013). On the use of Elo rating for adaptive assessment. *Studia Universitatis Babes-Bolyai, Informatica*, 58(1), 29-41.
- [15] Chen, C.M., Lee, H.M., & Chen, Y.H. (2005). Personalized e-Learning System Using Item Response Theory, *Computers & Education*, 44(3), 237-255. <https://doi.org/10.1016/j.compedu.2004.01.006>
- [16] Elo, A. E. (1978). *The rating of chess players past and present*, New York: Arco Publishing.
- [17] Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph*, No. 17.
- [18] Attali, Y. (2011). Immediate feedback and opportunity to revise answers: Application of a graded response IRT model. *Applied Psychological Measurement*, 35, 472-479.
- [19] Bergner, Y., Choi, I., & Castellano, K. E. (2019). Item Response Models for Multiple Attempts With Incomplete Data. *Journal of Educational Measurement*, 56(2), 415-436.
- [20] Forero, C. G., & Maydeu-Olivares, A. (2009). Estimation of IRT Graded Response Models: Limited Versus Full Information Methods. *Psychological Methods*, 14(3), 275-299.
- [21] Stephenson, A., and Sonas, J. (2019). R package "PlayerRatings". Retrieved from <https://CRAN.R-project.org/package=PlayerRatings>
- [22] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006, August). Yale: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 935-940).

Social Media Mining to Understand the Impact of Co-operative Education on Mental Health

Mohammad S. Parsa and Lukasz Golab
University of Waterloo
{mohammad.parsa,lgolab}@uwaterloo.ca

ABSTRACT

Co-operative education is a form of work-integrated learning that includes both classroom study terms and paid work experience. Research on co-operative education focuses on its benefits to students, employers, and academic institutions. In contrast, we study the impact of co-operative education on students' mental well-being. To do so, we mine social media content on the Reddit platform, which includes, among many other topics, discussion forums for major U.S. and Canadian colleges. Specifically, we perform topic modelling of discussions related to mental health and co-operative education. We find that students report feelings of self-doubt resulting from a competitive co-op job market, especially when placed in entry-level jobs that are not related to their academic programs, and anxiety due to job interviews, especially when they coincide with exams and other academic deadlines.

1. INTRODUCTION

Co-operative education (co-op) programs combine academic content with paid work experience. For example, students may alternate between classroom study terms and work-terms. Co-operative education programs, both at the undergraduate and graduate levels, have become popular as they offer practical work experience for students and a talent pipeline for employers [3, 26].

Prior work has examined the effect of co-operative education on students and employers. From a student point of view, studies have illustrated the impact of co-op on skill and career growth (see, e.g., [22, 13]). From an employer point of view, there has been work on understanding employers' expectations (see, e.g., [4, 16, 19]). On the other hand, there is less work on the effect of co-op on students' mental well-being, aside from small-scale studies of specific issues such as failing to obtain co-op employment (details in Section 2). This is, however, a pressing issue as recent work reports a rise in mental health problems among college students [1].

To fill this gap, we analyze social media to discover what students say about the impact of co-operative education on their well-being. Specifically, we perform topic modelling of U.S. and Canadian university discussion forums on the Reddit social media platform (reddit.com), followed by a detailed inspection of topics related to co-op.

In contrast to prior work based on surveys of small groups of students from a single institution, our study is based on a large dataset containing student-generated social media content from over 50 institutions, and is not limited to specific issues or students in specific circumstances. Furthermore, it has been recognized that the anonymity of social media makes it suitable for discussing sensitive issues. However, while there has been prior work on using social media such as Reddit and Twitter to understand mental health issues [7, 5, 18, 15, 14, 21, 8], including issues experienced specifically by students [1], these studies have not reported any issues related to co-operative education.

Our main findings are as follows. First, we find indications of self-doubt resulting from competition, specifically by students unable to secure highly-paid and popular co-op positions, and by students placed in entry-level jobs that are unrelated to their academic programs. Second, interviews for co-op positions appear to be causing anxiety: students fear being unprepared or unqualified, especially when interviews coincide with exams. These findings suggest actionable insights for academic institutions, including managing students' expectations and ensuring that co-op interviews do not conflict with academic deadlines.

2. RELATED WORK

In the context of social media mining, the closest work to ours is that of Bagroy et al. [1], which proposed a mental well-being index for college campuses. The index was computed by measuring the fraction of a given college's Reddit discussions that were related to mental health issues, as determined by a classifier. In a related study, Saha et al. [24] computed the fraction of these discussions that was classified as hate speech, and identified expressions of stress linked to exposure to hate speech. However, these studies did not report any issues related to co-op.

Next, we review related survey-driven studies. Drysdale and McBeath [12] surveyed 1970 students about psychological attributes such as hope, procrastination, self-efficacy, and study skills. They found that co-op students had lower anx-

Mohammad S. Parsa and Lukasz Golab "Social Media Mining to Understand the Impact of Co-operative Education on Mental Health" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 653 - 657

ity, a better attitude, better use of study aids, and better time management. Drewery et al. [10] surveyed 1989 co-op students and found that students who see a strong connection between the work term and their academic program are more likely to feel satisfied and perform well. Rowe [23] surveyed 29 researchers about neglected negative aspects of co-op. Some of the reported issues were related to mental health, e.g., depression of students unable to find co-op jobs or placed in jobs that are unrelated to their academic programs, and disconnect from campus life caused by alternating work and study terms. Cormier and Drewery [6] surveyed 82 students and found that those who did not find co-op employment reported negative feelings. On a similar note, Drewery et al. [11] tested two interventions, on 74 participants, to improve unemployed co-op students' well-being: a writing exercise and information about coping with stress. The first was found to be effective, but not the second. Finally, Deziel et al. [9] surveyed 312 students about their mental health and found that it is related to academic and demographic factors such as program, year of study, and gender. However, the effect of co-op was not considered.

3. DATA AND METHODS

3.1 Data Collection and Pre-Processing

Reddit is an online social media platform divided into over 100,000 user-created discussion communities referred to as *subreddits*. A subreddit contains a number of *posts* that initiate discussions, and a post is followed by (zero or more) *comments*. Subreddit names begin with “r/” and are indicative of the content. For example, r/Fitness contains discussions of fitness and physical exercise, r/StarWars is a forum for fans of Star Wars movies, etc. As of 2019, there are over 400 million users on Reddit. Each user has a Reddit ID, but is not required to reveal any personal information.

Previous work [1] has identified the subreddits corresponding to top U.S. colleges according to U.S. News¹. We also use these subreddits in our analysis, listed in Table 1. Additionally, we collected the subreddits corresponding to top Canadian universities according to McLean’s Magazine², listed in Table 2. We downloaded these subreddits (posts and comments) from a publicly accessible database on Google Big Query³, spanning from September 2015 to September 2019. The sizes of each studied subreddit are shown in Table 1 and 2, in the “before” columns; the numbers in the “after” columns refer to content relevant to mental health and co-operative education, as determined by our filtering methods described in Section 3.2.

Next, we perform standard text pre-processing. Following previous work on Reddit data mining [17], we remove posts and comments with fewer than 256 or more than 4096 characters: short ones are unlikely to be meaningful (and may instead correspond to URLs), while long ones may mention more than one topic. We also remove stopwords and we lemmatize the remaining words (i.e., we group together all the *inflected* forms of a word) using the Python NLTK parser.

¹<https://www.usnews.com/best-colleges/rankings/national-universities>

²<https://www.macleans.ca/education/university-rankings-2020-canadas-top-comprehensive-schools/>

³<https://cloud.google.com/bigquery>

Table 1: U.S. academic subreddits: number of posts and comments before and after processing.

Subreddits	Posts		Comments	
	before	after	before	after
r/UIUC	5893	423	27258	1864
r/rutgers	4062	263	12858	913
r/UMD	2861	198	10748	916
r/UCSD	2638	163	10991	771
r/Purdue	2408	183	10540	883
r/berkeley	2254	170	15946	970
r/UTAustin	2134	133	8744	507
r/utdallas	1974	146	6476	524
r/gatech	18623	305	14605	1386
r/Cornell	1718	75	5865	453
r/udub	1571	97	7422	456
r/uofm	1550	99	6254	546
r/SBU	1450	90	4367	246
r/rit	1363	96	7080	684
r/UWMadison	1322	93	5040	343
r/RPI	1207	91	7119	571
r/SJSU	1187	51	3955	306
r/nyu	1146	82	2975	169
r/PennStateUniversity	1134	85	5255	369
r/NCSU	1110	58	3679	293
r/msu	1074	52	4191	316
r/UGA	1026	69	3511	278
r/USC	931	51	2851	197
r/UVA	616	49	2273	116
r/uichicago	532	37	1178	93
r/UNCCharlotte	512	32	1635	94
r/stanford	510	42	1416	112
r/UPenn	491	35	1276	68
r/columbia	411	30	1428	55
r/cmu	333	25	1320	118
r/Baruch	324	19	796	57
r/IndianaUniversity	320	25	1311	79
r/mit	316	25	1487	121
r/UMBC	286	11	822	59
r/Harvard	241	18	1081	73
r/BrownU	219	14	603	28
r/byu	198	20	1404	75
r/duke	187	10	502	19
r/UNC	184	9	416	21
r/washu	179	15	622	30
r/Vanderbilt	156	9	334	16
r/bostoncollege	96	3	315	9
r/Caltech	77	11	232	20
Total	66824	3512	208181	15224

3.2 Content Filtering

Academic subreddits discuss a variety of topics related to the corresponding college, such as admissions, academics and campus events. Thus, the next step is to *filter* the data and identify discussions that are relevant to our analysis, namely those which 1) are related to mental health, and 2) are related to co-operative education.

First, we apply a *classifier* that predicts whether a post or a comment is likely to be related to mental health. We use the logistic regression classifier from Bagroy et al. [1], which was originally used to compute the percentage of discussions on academic subreddits that are related to mental health. This classifier was trained by considering all posts on the subreddit r/mentalhealth to be mental-health-related and all posts on control subreddits (among them r/food, r/technology, and the FAQ forum r/AskReddit) to be unrelated.

Next, we only retain posts and comments that appear re-

Table 2: Canadian academic subreddits: number of posts and comments before and after processing.

Subreddits	Posts		Comments	
	before	after	before	after
r/uwaterloo	8912	1836	43382	5215
r/UofT	7895	588	32929	2490
r/UBC	3577	406	20504	1485
r/uAlberta	2766	146	8968	576
r/yorku	2612	182	9877	531
r/mcgill	2603	171	10517	635
r/Concordia	1643	129	4042	286
r/uwo	1599	112	6167	401
r/ryerson	1383	82	4018	374
r/CarletonU	1320	134	4988	473
r/McMaster	928	63	2354	218
r/queensuniversity	763	35	2452	163
r/uvic	665	41	2400	181
r/wlu	458	38	1159	114
r/uoguelph	399	29	960	99
r/Dalhousie	293	24	667	39
r/umanitoba	165	23	354	18
r/brocku	103	11	220	22
r/memorialuniversity	86	5	116	15
r/usask	74	4	157	11
r/uottawa	35	5	27	2
r/UdeM	22	2	101	4
r/University_Of_Regina	21	3	30	3
r/lakeheadu	20	0	47	4
r/uileth	17	0	18	3
r/laurentian	14	0	21	0
r/AcadiaU	14	1	44	1
Total	38387	4070	156519	13363

lated to co-op, and we do this by only keeping those which contain at least one of the following co-op related terms: “coop”, “interview”, “resume”, “workterm”, and “intern”. Note that we lemmatized the words during pre-processing, so “interview” also captures similar words such as “interviewer” or “interviewing”.

3.3 Topic Modelling

We then apply topic modelling to the posts and comments that passed the above mental health and co-op filters. First, we vectorize the comments and posts in a standard way. For each post or comment, the i th entry of its vector corresponds to the Term Frequency - Inverse Document Frequency (TF-IDF) of the i th word. We compute the TF-IDF score of a given word for a given post or comment as follows: we divide the number of times the word appears in the given post or comment (TF) by the fraction of total posts and comments that contain at least one occurrence of this word (DF). TF-IDF is frequently used when vectorizing text as it takes into account both the uniqueness of a word in the entire dataset and the importance of the word to the specific document (in our case, the specific post or comment).

Next, we run the Non-negative Matrix Factorization (NMF) topic modelling algorithm [27] on the vectorized posts and comments. NMF clusters the data into topics and produces a list of representative terms called *topic descriptors* for each topic. Each such term has a “representativeness” score, and we select the top-10 highest-scoring terms for each topic. Additionally, for each topic, we report the top-10 most frequent word n -grams (for n up to three, i.e., sequences of up to three consecutive words) within the posts or comments

belonging to the given topic.

NMF requires the number of topics as input. To select an appropriate number of topics, we ran NMF to produce between 2 and 100 topics, and computed the *coherence* [20] of each output (higher is better). We obtained the highest coherence for ten topics.

Another issue with NMF is that despite our text pre-processing, some topic descriptors were uninformative. Following prior work on topic modelling [17, 25, 2], we repeatedly remove uninformative terms from the posts and comments and re-run NMF until the topic descriptors no longer contain any uninformative terms. After two iterations, all the top-10 descriptors became informative.

Finally, we extract issues affecting students from the NMF topic descriptors, the frequent n -grams, and a manual inspection of a 5% sample of posts and comments assigned to each topic.

4. RESULTS

Table 3 shows the topic modeling results for posts and comments related to both mental health and co-op, including topic descriptors, a sample of frequent n -grams, and the percentage of content assigned to each topics. After inspecting these results, plus a sample of posts and comments assigned to each of the ten topics, we manually group the topics into issues, as shown in Table 4 (where we also point out which topics from Table 3 describe which issue).

Topics 1, 2, 4, 9 and 10 cover over 60 percent of the content and appear related to competition, specifically the competitive nature of the co-op job market. Upon manual inspection of a sample of posts and comments, we found that students express self-doubt and feelings of inadequacy when unable to secure a desirable co-op job, especially when one’s classmates and friends are able to obtain such jobs. There were also some discussions about choosing a good co-op program that enables interesting and highly-paid co-op job opportunities, concerns over not having enough experience to qualify for these desirable jobs, and the stress of maintaining a high GPA to qualify for or remain in such programs. Notably, many of the posts and comments related to competition referred to technology and software roles, as well as large technology employers such as Facebook and Google. This is likely due to the fact that co-operative programs are mainly in science and engineering.

Next, topics 3 and 5 are about questions students ask about co-op programs. This includes general questions related to admissions, and specific questions such as how to write a work report.

Topic 6 describes issues with interviews. Many posts and comments belonging to this topic referred to interviews for co-op jobs being stressful, especially because they often coincide with exams and other academic deadlines, and because interview processes for software positions may include lengthy programming tests. Students also reported feelings of uncertainty about how to prepare for interviews, how to acquire required skills, and what to expect. Additionally, some students reported anxiety after an interview while

Table 3: Topic modeling output for co-op related posts and comments

Topic descriptors	Frequent n-grams	%
1 work, time, people, like, make, want, hard, day, need, know	'work hard', 'people work', 'school work'	29.9
2 job, apply, degree, graduate, student, want, like, people, look, pay	'apply job', 'job market'	14.1
3 project, code, like, use, course, make, time, start, personal, create	'work project', 'start project', 'personal project'	12.2
4 experience, internship, year, co-op, school, summer, gpa, company, graduate, program	'work experience', 'grad school', 'work hard'	9.5
5 class, easy, semester, final, hard, pretty, time, exam, course, lecture	'class work', 'final project', 'work time'	7.8
6 resume, interview, look, company, ask, skill, apply, recruiter, employer, page	'work experience', 'career fair', 'cover letter'	7
7 lab, research, professor, prof, student, grad, undergrad, paper, ask, email	'research project', 'work lab', 'grad student'	5.6
8 group, member, people, meet, person, presentation, individual, make, facebook, fb	'group project', 'work group', 'class group'	5.3
9 team, game, member, join, play, club, engineer, player, people, design	'project team', 'work project', 'team work'	5
10 letter, cover, apply, write, application, position, make, generic, company, tailor	'cover letter', 'resume cover', 'resume cover letter'	3.6

Table 4: Issues extracted from co-op related posts and comments

Topics	Issue	Description	%
1,2,4, 9,10	Competition	E.g. not qualifying for a desired co-op job	62.1
3,5	Questions	About co-op programs (e.g., seeking clarification when instructions are not clear enough)	20
6	Interviews	E.g., not knowing what to expect or how to prepare	7
7	Research opportunities	Not directly related to co-op	5.6
8	Group projects	Not directly related to co-op	5.3

waiting to find out if they have been hired.

Finally, topics 7 and 8 are not directly related to the effect of co-op on students' well-being; they instead refer mainly to research opportunities and participation in group projects during internships.

5. DISCUSSION AND CONCLUSIONS

By performing topic modelling on subreddits corresponding to U.S. and Canadian universities, we obtained the following insights into the impact of co-operative education on students' well-being.

1. **Competition** for internships, especially in the software and information technology fields, is a frequently discussed negative aspect of co-operative education. Prior work has observed that co-op unemployment can lead to mental well-being issues [6, 11]. However, our results further indicate that not securing a desirable, high-paying, challenging, and relevant employment can be a source of stress, self-doubt, and disappointment. This is especially true if one's friends and classmates are able to secure desirable jobs that are directly related their programs of study.
2. **Co-op interviews** are a source of stress for several reasons. First, students fear being unprepared or un-

qualified, especially when competing for sought-after jobs. Second, interviews often coincide with midterm examinations and other academic deadlines, meaning that students may have to choose between preparing for interviews (including preparing for programming tests) and coursework. Previous work has argued that co-operative education research should consider work-related variables in addition to education-related ones; these work-related variables include skills, job satisfaction, performance assessments, and selection interviews [23]. Our findings on co-op interviews align with this suggestion, providing data-driven evidence of another source of anxiety for co-op students.

3. As reported in previous work [23], we also found some reports of **loneliness during workterms**. Additionally, **moving and finding a place to live** during a workterm can be a source of stress.

Our findings suggest actionable insights for academic institutions and students. First, it is important to manage co-op students' expectations. For example, universities may want to offer workshops that explain the competitive nature of the co-op process and help students find jobs they qualify for. Junior students, specifically, should keep in mind that they may not immediately qualify for the sought-after positions secured by their senior colleagues. Additionally, these workshops should provide advice on interview preparation, coping with frequent moving, and finding short-term living arrangements during internships. Second, co-op interviews should not be scheduled during peak academic times. Having more time to prepare, especially for software interviews with programming tests, may reduce anxiety.

One limitation of this study is that it only reflects the opinions of students who are active on Reddit. Nevertheless, our findings can be used as a starting point for additional focused research. In future work, we plan to survey students to confirm our findings about the competitive nature of the co-op job market. Additionally, we will analyze course discussion forums to further investigate the impact of co-op interviews on class schedules and academic deadlines.

6. REFERENCES

- [1] S. Bagroy, P. Kumaraguru, and M. De Choudhury. A social media based index of mental well-being in college campuses. In *Proc. CHI Conference on Human factors in Computing Systems*, pages 1634–1646, 2017.
- [2] A. Blanchard. Understanding and customizing stopword lists for enhanced patent mapping. *World Patent Information*, 29(4):308–316, 2007.
- [3] L. A. Braunstein and W. A. Stull. Employer benefits of, and attitudes toward postsecondary cooperative education. *Journal of Cooperative Education*, 36(1):7, 2001.
- [4] S. Chopra and L. Golab. Job description mining to understand work-integrated learning. In *Proc. Int. Conf. on Educational Data Mining*, pages 32–43, 2018.
- [5] G. Coppersmith, M. Dredze, and C. Harman. Quantifying mental health signals in twitter. In *Proc. Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 51–60, 2014.
- [6] L. Cormier and D. Drewery. Examining the effect of co-op non-employment and rejection sensitivity on subjective well-being. *Asia-Pacific Journal of Cooperative Education*, 18(3):213–224, 2017.
- [7] M. De Choudhury and S. De. Mental health discourse on reddit: Self-disclosure, social support, and anonymity. In *Proc. Int. AAAI Conf. on Weblogs and Social Media*, 2014.
- [8] M. De Choudhury, S. S. Sharma, T. Logar, W. Eekhout, and R. C. Nielsen. Gender and cross-cultural differences in social media disclosures of mental illness. In *Proc. ACM Conf. on Computer Supported Cooperative Work and Social Computing*, pages 353–369, 2017.
- [9] M. Deziel, D. Olawo, L. Truchon, and L. Golab. Analyzing the mental health of engineering students using classification and regression. In *Proc. Int. Conf. on Educational Data Mining*, pages 228–231, 2013.
- [10] D. Drewery, T. J. Pretti, and S. Barclay. Examining the effects of perceived relevance and work-related subjective well-being on individual performance for co-op students. *Asia-Pacific Journal of Cooperative Education*, 17(2):119–134, 2016.
- [11] D. W. Drewery, L. A. Cormier, T. J. Prettti, and D. Church. Improving unmatched co-op students’ emotional wellbeing: Test of two brief interventions. *International Journal of Work-Integrated Learning*, 20(1):43–53, 2019.
- [12] M. T. Drysdale and M. McBeath. Exploring hope, self-efficacy, procrastination, and study skills between cooperative and non-cooperative education students. *Asia-Pacific Journal of Cooperative Education*, 15(1):69–79, 2014.
- [13] J. Gault, J. Redington, and T. Schlager. Undergraduate business internships and career success: are they related? *Journal of Marketing Education*, 22(1):45–53, 2000.
- [14] G. Gkotsis, A. Oellrich, T. Hubbard, R. Dobson, M. Liakata, S. Velupillai, and R. Dutta. The language of mental health problems in social media. In *Proc. Workshop on Computational Linguistics and Clinical Psychology*, pages 63–73, 2016.
- [15] S. C. Guntuku, D. B. Yaden, M. L. Kern, L. H. Ungar, and J. C. Eichstaedt. Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, 18:43–49, 2017.
- [16] D. Hodges and N. Burchell. Business graduate competencies: Employers’ views on importance and performance. *International Journal of Work-Integrated Learning*, 4(2):16, 2003.
- [17] A. Khan and L. Golab. Reddit mining to understand gendered movements. In *Proc. EDBT Workshop on Data Analytics Solutions for Real-Life Applications*, 2020.
- [18] C. McClellan, M. M. Ali, R. Mutter, L. Kroutil, and J. Landwehr. Using social media to monitor mental health discussions- evidence from twitter. *Journal of the American Medical Informatics Association*, 24(3):496–502, 2017.
- [19] C. Nevison, L. Cormier, J. Pretti, and D. Drewery. The influence of values on supervisors’ satisfaction with co-op student employees. *International Journal of Work-Integrated Learning*, 19(1):1–11, 2018.
- [20] D. O’callaghan, D. Greene, J. Carthy, and P. Cunningham. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*, 42(13):5645–5657, 2015.
- [21] U. Pavalanathan and M. De Choudhury. Identity management and mental health discourse in social media. In *Proc. Int. Conf. on World Wide Web*, pages 315–321, 2015.
- [22] E. Ralph, K. Walker, and R. Wimmer. Practicum-education experiences: Post-interns’ views. *International Journal of Engineering Education*, 25:122–130, 01 2009.
- [23] P. M. Rowe. Researchers’ reflections on what is missing from work-integrated learning research. *Asia-Pacific Journal of Cooperative Education*, 16(2):101–107, 2015.
- [24] K. Saha, E. Chandrasekharan, and M. De Choudhury. Prevalence and psychological effects of hateful speech in online college communities. In *Proc. ACM Conference on Web Science*, pages 255–264, 2019.
- [25] A. Toulis and L. Golab. Social media mining to understand public mental health. In *VLDB Workshop on Data Management and Analytics for Medicine and Healthcare*, pages 55–70, 2017.
- [26] G. Van Gyn, J. Cutt, M. Loken, and F. Ricks. Investigating the educational benefits of cooperative education: A longitudinal study. *Science*, 97(180):277, 1997.
- [27] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proc. Int. ACM SIGIR Conf. on Research and Development in Informaion Retrieval*, pages 267–273, 2003.

Towards Temporality-Sensitive Recurrent Neural Networks through Enriched Traces

Thomas Sergent
Sorbonne Université, CNRS,
LIP6, F-75005 Paris, France
Lalilo, Paris, France
thomas.sergent@lip6.fr

François Bouchet
Sorbonne Université, CNRS,
LIP6, F-75005 Paris, France
francois.bouchet@lip6.fr

Thibault Carron
Sorbonne Université, CNRS,
LIP6, F-75005 Paris, France
thibault.carron@lip6.fr

ABSTRACT

Educational traces are distinctive compared to the usual data a recurrent neural network encounters: there is a difference between two consecutive educational traces generated by a same learner if they are separated by 2 minutes or 2 months. Indeed, in the latter case, the learner who generated the trace may have forgotten the associated skill, which is less likely in the former case. Recurrent Neural Networks have seen a surge of popularity in the recent few years thanks to Deep Knowledge Tracing. While the focus has mostly been on the network architecture, we propose here a novel framework where traces are enriched with information relative to the temporality before they are used to train the network, and assess the performance on two datasets (Lalilo and ASSISTments 2012), which is not improved by this approach.

Keywords

Educational Data Mining, Neural Networks, Trace enrichment, Temporality

1. INTRODUCTION

1.1 Modelling student learning

As reminded by Choffin et al. [3], there are two main approaches to model students' learning: knowledge tracing and factor analysis.

On the one hand, knowledge tracing approaches model students' learning over time by nature by taking into account the sequential order of traces. Historically, these approaches started with Hidden Markov Models (HMMs) which were particularly used for Bayesian Knowledge Tracing (BKT) [4]. More recently, Deep Knowledge Tracing (DKT) appeared and spread partly thanks to increases in computing power [13]. The key idea is to model skills mastered by the students using Recurrent Neural Networks (RNNs).

On the other hand, Factor Analysis models have been developed since the 1950s (cf. [15] for a recent synthesis). They

rely on the idea of making explicit the factors that can have an impact on students' success on a given exercise. Training the model then consists in computing the weight of those various factors in success.

1.2 How Recurrent Neural Networks are usually used

Thanks to massive increase in GPUs computing power, research in artificial neural networks and deep learning has developed at a fast pace over the past decade [10]. In the EDM community, its first use came with DKT in 2014, which uses RNNs in order to continuously model students' learning over time. Since then, several variations of RNNs have been created to better model students learning, such as Dynamic Key-Value Memory Networks for Knowledge Tracing (DKVMN) [17] and more recently Deep Hierarchical Knowledge Tracing (DHKT) [16] and Knowledge Query Network [11]. However all those alternative models only work on trying to adapt the structure itself of the RNN.

Outside of the EDM community, today some of the main uses of RNNs include natural language translation, speech recognition and time series forecasting. Those three uses have one common point: the distance between two successive data is always the same (a single space between two words of a text, 25ms between two audio samples in speech recognition or a same time difference between two points in time series). Nonetheless, this property is usually not true on problems datasets generated by students using various learning platforms or intelligent tutoring systems (ITSs) which are commonly considered in the EDM community. Indeed, in this context two consecutive log entries could be separated by two minutes (for two exercises done during the same learning session) or two months (if the student stops using the learning platform for a while). And when this time is long, it is likely that the student has either significantly progressed on that skill (through work outside of the system, e.g. class work) or on the contrary they may have forgotten some previously mastered skills.

1.3 Beyond sequentiality: temporality

But there are reasons to think that taking into account time data can significantly improve success prediction for an exercise. The importance of modeling temporal aspects in analyses of learning has been well-established in the EDM and LAK communities [9]. In sequence mining approaches for instance, authors tend to take it into account by allowing gaps between actions [8] and/or with feature engineering

Thomas Sergent, François Bouchet and Thibault Carron "Towards Temporality-Sensitive Recurrent Neural Networks through Enriched Traces" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 658 - 661

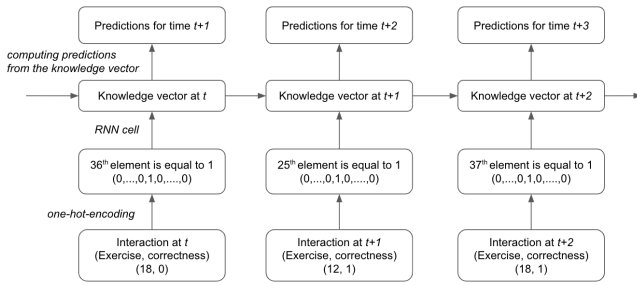


Figure 1: DKT structure. Adapted from [13]

by integrating this aspect in the actions themselves (short vs. long actions) [2]. The DAS3H model [3], which does not use RNNs, uses time windows of various durations to try to characterize the slopes of the forgetting curves for each skill. In deep learning approaches, Nagatani et al. [12] slightly modify the RNN structure to use exercise counts and time gaps between two similar exercises as additional inputs and are able to increase the AUC (Area Under ROC Curve) of predictions. In the medical field, a similar study to predict patient follow-ups benefited from a modification in the LSTM cell making it time-aware [1]. So likewise here we want to investigate ways to use the traces temporality and whether this could improve the quality of the predictions.

2. METHOD

2.1 Deep Knowledge Tracing in a nutshell

As mentioned before, the principle of DKT and its variants relies on an RNN whose weights characterize how a student's skills mastery evolves after an interaction with a learning system, such as an ITS. An interaction with the system is represented as a couple {exercise, answer}. So if there are K exercises with a boolean answer, $2K$ interactions are possible. In order to facilitate calculations in RNNs, they are usually one-hot-encoded using a binary vector of $2K$ values. For a sequence of N values, each interaction gets sequentially through the RNN. At a time t , the t -th interaction goes through the cell of the RNN thus providing the new vector representing the estimated knowledge of the student at that time. This vector can then be used to predict success on a given exercise at time $t + 1$ (cf. Figure 1). Training a RNN thus corresponds to learning the transitions between a given student's knowledge vectors. We can notice that nowhere the temporal distance (or time gap) between two inputs is considered (cf. Figure 2 top), and as far as the authors know, no knowledge tracing algorithms are currently considering it.

2.2 Our proposal

Usually, enriching the traces consists in feature engineering and tends to be presented as an alternative to Recurrent Neural Networks [6]. [12] and [1] added temporality by modifying the structure of the neural network to include temporal information. Here we try a different approach by inserting new traces in the dataset, doing meta feature-engineering to be used by a Recurring Neural Network. Our idea consists in considering all traces from students as a sequence with missing values when there are no new trace for a given period of time. In cases like this, one can usually infer the

Dataset	% of traces spaced by	
	> 7 days	> 30 days
ASSISTments12	3.1	0.6
ASSISTments17	1.1	0.4
Algebra I 2005-2006	0.3	0.02
Bridge to Algebra 2006-2007	0.02	0.003
Lalilo	2.0	0.4

Table 1: Dataset traces spread

missing values [7] by (1) adding data whose values are equal to the average of previous data or (2) adding again the same data that was last added. However, neither of these two approaches can be applied here. First, they are typically used for time series where only some variables are missing at time t but not all of them, whereas here it is equivalent to having all variables missing at time t . Moreover, averaging previous interactions does not make sense mathematically speaking. Finally, adding again the same interaction that was last added would not take into account the fact the student may have been progressing or forgetting during the time in which they were not using the learning platform.

Our proposal thus consists in adding traces (further on referred to as "artificial traces") at a regular predefined static time interval when students are not using the learning platform. The underlying hypothesis is that the RNN will be able to interpret those as time passing by. Those artificial traces are added as exercise $K + 1$ (knowing there are only K exercises initially). Thus if we add traces every month without any exercise done, after 3 months without use, 2 artificial traces will have been added (cf. Figure 2 middle). In a similar scenario, if traces are added every week without any exercise done, 11 artificial traces will have been added (cf. Figure 2 bottom).

Adding those traces results in modifying the student's knowledge vector after each exercise $K + 1$. After a given time without any new exercise done, the predictions corresponding to the probability of success will therefore differ from the ones without any artificial trace. When the RNN learns the meaning of that $K + 1$ -th exercise, it could lead to an improvement of the predictions.

In order to keep the initial tuple structure, we also add an arbitrary correctness of 0, which is not used practically to train the network.

3. EXPERIMENTS

3.1 Experimental setting

We computed the number of traces spaced by more than 7 and 30 days in a number of classical datasets : ASSISTments12 [5], ASSISTments17, Algebra I 2005-2006 and Bridge to Algebra 2006-2007. The two latter datasets stem from the KDD Cup 2010 EDM Challenge [14]. We were also able to get a dataset from Lalilo which is a web-app fostering literacy for K-2 (Table 1). In order to evaluate the performance of the traces enrichment, we have trained DKT and DHKT algorithms on the two datasets that had the highest spread in traces: ASSISTments12 (assist12) and Lalilo. Their main characteristics are summarized in Table 2.

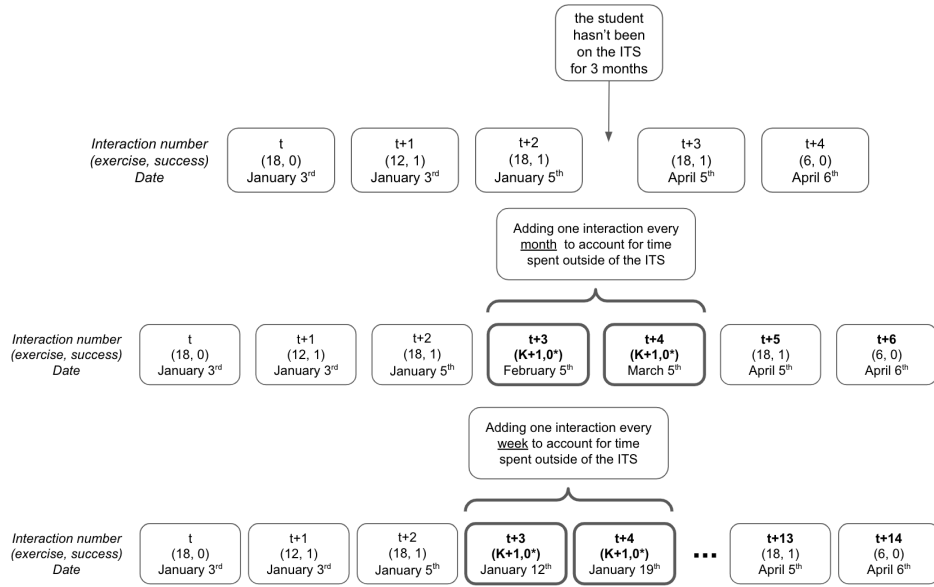


Figure 2: Traces fed to the network before enrichment (top), after *monthly* temporal enrichment (middle), or after *weekly* temporal enrichment (bottom)

Dataset	Users	Items	Skills	Interactions	Median length
lalilo	58,585	3,439	16	4,418,190	46
assist12	29,018	53,086	265	2,711,602	49

Table 2: Datasets characteristics

A key question with our approach relates to the frequency of use to add artificial traces. Indeed if the frequency is too high, it is likely that the artificial traces would disturb the RNN learning. Conversely, if the frequency is too low, it won't capture precisely the elapsed time. Therefore we compared the impact of various frequencies of artificial traces addition.

Our models have been implemented in Python and PyTorch for the Deep Learning aspects and the corresponding code is available online on GitHub¹. Following Choffin et al. [3], we removed users for whom the number of interactions was less than 10 and interactions with NaN skills. We randomly sample randomly training (80%) and testing (20%) sets and give results on the testing set. We average on five different seeds and give standard deviation.

3.2 Results and analysis

A synthesis of the results can be found in the Tables 3 and 4. We use AUC to evaluate the performance of the models. No significant improvement in the predictions appear, and even with a high frequency of added artificial traces (daily), there is no significant degradation either. Several hypothesis could explain this lack of impact. It is possible that in those datasets, students are not progressing or regressing significantly between two moments when they use ASSISTments. A lack of differences would also be likely to be observed if long gaps without usage are unlikely. For example, if most

¹<https://github.com/thosgt/kt-algos>

Model	Added trace frequency (# days)	AUC (std dev)
DKT	None	0.734 (0.004)
DKT	1	0.735 (0.003)
DKT	7	0.734 (0.002)
DKT	14	0.734 (0.002)
DKT	30	0.734 (0.005)
DHKT	None	0.771 (0.002)
DHKT	1	0.770 (0.002)
DHKT	7	0.771 (0.005)
DHKT	14	0.771 (0.003)
DHKT	30	0.770 (0.002)

Table 3: Performance comparison on the ASSISTments12 dataset

students do 50 exercises over a few days, then stop using the system for 2 months, and use it again intensively for a week, the only exercises impacted would be the ones right after the gap of two months, i.e. only a small percentage of exercises overall. It is thus also possible that other datasets would be more sensitive to measure the impact of this artificial traces addition.

4. CONCLUSION AND PERSPECTIVES

We proposed here a framework to enrich learning traces to train recurrent neural networks. This enrichment which consists in adding artificial traces allows to add a temporality aspect into traces which normally only take into account se-

Model	Added trace frequency (# days)	AUC (std dev)
DKT	None	0.685 (0.001)
DKT	1	0.685 (0.003)
DKT	7	0.685 (0.002)
DKT	14	0.684 (0.003)
DKT	30	0.685 (0.002)
DHKT	None	0.701 (0.002)
DHKT	1	0.700 (0.002)
DHKT	7	0.701 (0.001)
DHKT	14	0.702 (0.003)
DHKT	30	0.700 (0.001)

Table 4: Performance comparison on the Lalilo dataset

quentiality. Unfortunately, ASSISTments 2012 and Lalilo datasets did not allow us to reveal a significant impact of our approach, but we have reasons to believe these particular datasets were not the most appropriate to measure a significant difference in the performance of prediction. Our future works thus involve (1) focusing on a population of students who has a scarce use of a learning platform over a large period of time (several months or years), (2) focusing on the impact of this algorithm over prediction specifically on exercises done right after a large time gap (during which the student may have learned or forgotten things), and (3) identifying learning platforms that teaches skills that are maybe easier to forget over time (e.g. vocabulary in a foreign language), or finding already existing datasets coming from such a platform.

5. ACKNOWLEDGMENTS

We would like to thank the reviewers for their thoughtful comments and efforts towards improving our paper. This work is funded by Lalilo.

6. REFERENCES

- [1] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou. Patient Subtyping via Time-Aware LSTM Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 65–74, Halifax, NS, Canada, Aug. 2017. Association for Computing Machinery.
- [2] F. Bouchet, R. Azevedo, J. S. Kinnebrew, and G. Biswas. Identifying Students' Characteristic Learning Behaviors in an Intelligent Tutoring System Fostering Self-Regulated Learning. *International Educational Data Mining Society*, 2012.
- [3] B. Choffin, F. Popineau, Y. Bourda, and J.-J. Vie. DAS3H: Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills. *EDM*, 2019.
- [4] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, Dec. 1994.
- [5] M. Feng, N. Heffernan, and K. Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction*, 19(3):243–266, Aug. 2009.
- [6] Y. Jiang, N. Bosch, R. S. Baker, L. Paquette, J. Ocumpaugh, J. M. A. L. Andres, A. L. Moore, and G. Biswas. Expert Feature-Engineering vs. Deep Neural Networks: Which Is Better for Sensor-Free Affect Detection? In *Artificial Intelligence in Education*, volume 10947, pages 198–211. Springer International Publishing, Cham, 2018.
- [7] Y. J. Kim and M. Chi. Temporal Belief Memory: Imputing Missing Data during RNN Training. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 2326–2332, Stockholm, Sweden, July 2018. International Joint Conferences on Artificial Intelligence Organization.
- [8] J. S. Kinnebrew and G. Biswas. *Identifying Learning Behaviors by Contextualizing Differential Sequence Mining with Action Features and Performance Evolution*. International Educational Data Mining Society, June 2012.
- [9] S. Knight, A. F. Wise, B. Chen, and B. H. Cheng. It's about time: 4th international workshop on temporal analyses of learning data. In *Proceedings of the 5th International Conference on Learning Analytics and Knowledge, LAK 2015*, pages 388–389. Association for Computing Machinery, Mar. 2015.
- [10] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.
- [11] J. Lee and D.-Y. Yeung. Knowledge Query Network for Knowledge Tracing: How Knowledge Interacts with Skills. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge, LAK19*, pages 491–500, Tempe, AZ, USA, Mar. 2019. Association for Computing Machinery.
- [12] K. Nagatani, Q. Zhang, M. Sato, Y.-Y. Chen, F. Chen, and T. Ohkuma. Augmenting Knowledge Tracing by Considering Forgetting Behavior. In *The World Wide Web Conference, WWW '19*, pages 3101–3107, San Francisco, CA, USA, May 2019. Association for Computing Machinery.
- [13] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep Knowledge Tracing. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 505–513. Curran Associates, Inc., 2015. 00208.
- [14] J. Stamper, A. Niculescu-Mizil, S. Ritter, G. Gordon, and K. Koedinger. Algebra I 2005-2006 and Bridge to Algebra 2006-2007. Development data sets from KDD Cup 2010 Educational Data Mining Challenge. Find them at <http://pslcdatashop.web.cmu.edu/KDDCup/>.
- [15] J.-J. Vie and H. Kashima. Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):750–757, July 2019.
- [16] T. Wang, F. Ma, and J. Gao. Deep Hierarchical Knowledge Tracing. In *EDM*, 2019.
- [17] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic Key-Value Memory Networks for Knowledge Tracing. In *Proceedings of the 26th International Conference on World Wide Web, WWW '17*, pages 765–774, Perth, Australia, Apr. 2017. International World Wide Web Conferences Steering Committee.

Predicting and Understanding Success in an Innovation-Based Learning Course

Lauren Singelmann, Enrique Alvarez, Ellen Swartz, Ryan Striker, Mary Pearson, Dan Ewert
North Dakota State University
Lauren.N.Singelmann@ndsu.edu

ABSTRACT

In order to keep up with the rising demand for new and innovative solutions in an evolving world, an even greater importance is being placed on training engineers that can tackle big problems. However, the process of teaching engineering students to be innovative is not straightforward. There are multiple ways to demonstrate innovation and problem-solving abilities, meaning traditional educational data mining methods aren't always appropriate. To better understand the process of problem-solving and innovation, this work collected data from students working on innovation projects within a course and determined appropriate ways to gain information and insight from the data. Students wrote and categorized learning objectives in an online portal, which generated log data when they created, updated, and completed personal learning objectives and corresponding deliverables. Classification models that were both robust (ROC AUC > .95) and interpretable were applied to both the language used in the objectives and the quantifiable features such as number of objectives, time of completing certain milestones, and number of deletions and edits. By extracting the most significant features, we are able to see which variables are most likely to lead to student success in innovation-based learning. This would aid instructors in offering impactful support to students or eventually lead to an online tutoring system. The conducted analysis will help students develop and grow throughout the innovation process in this course or in other open-ended problem-solving environments.

Keywords

Classification, open-ended learning, innovation, problem-solving

1. INTRODUCTION

Thomas Friedman describes the current era as the *Age of Accelerations*, the time at which technology, the climate, and globalization are all evolving at a rate like we've never seen before [5]. As these areas progress, engineers need to be

able to identify and solve problems more quickly and effectively than ever before. ABET [1], the National Academies of Engineering [10], and experts in both engineering and education [12] all stress the growing importance for training engineers that can use their problem-solving skills to create new and innovative solutions. This work explores how students work on these skills and solve real-world problems in an Innovation-Based Learning (IBL) course. IBL students apply their content knowledge and skills to work on a real-world project with the goal of creating value external to the class. For example, successful students have presented their work at conferences, published papers, participated in invited outreach activities, or submitted invention disclosures. Students are required to write their own learning objectives and show evidence of work. However, because there are so many possible approaches to the course, predicting and understanding student success can be challenging.

In order to better understand what makes students successful in this type of course, data were collected from the online learning portal from the class. Because of the open-ended nature of the course, classification and knowledge discovery can be challenging. We wanted to build classifier models that were robust, but also interpretable in order to better predict and support future students in IBL-style courses. Therefore, three main research questions were explored:

RQ1: What feature sets and models work best for IBL data?

RQ2: How early can student success be predicted?

RQ3: What features are most likely to differentiate between top-performers and lower-performers?

Finding answers to all of these questions can help guide instruction in the course and potential development of an online tutoring system for innovation-based learning courses and other open-ended problem-solving environments. This paper will present literature about open-ended learning environments, give details about the course and data collected, elaborate on how the research questions will be tested, and share results and takeaways.

2. OPEN-ENDED LEARNING

Open-ended learning is a pedagogical approach that allows students to use their own motivations and approach to learn about the world [8]. Students are taking part in authentic problem-solving, practicing metacognition, and creating unique pathways through their learning. Examples of open-ended learning environments include computer pro-

Lauren Singelmann, Enrique Alvarez, Ellen Swartz, Ryan Striker, Mary Pearson and Dan Ewert "Predicting and Understanding Success in an Innovation-Based Learning Course" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 662 - 666

programming exercises, project-based learning, and inquiry-based learning. Educational research exists about the potential benefits of these experiences [11], but there are still gaps in understanding about how students progress through open-ended learning environments. Educational data mining (EDM) has shown great potential in being able to help shed light on student trajectories and habits within open-ended learning environments [20].

EDM has shown preliminary successes in open-ended learning environments such as learning computer programming [9], project-based learning courses [17], online tutoring platforms [3,4], learning-by-teaching platforms [6], and language tutoring systems [13]. Continuing to make strides in understanding learning in open-ended environments is imperative because it will be an important step in implementing evidence-based practices and assessment in these environments.

3. METHODS

3.1 Cardiovascular Engineering Course

Data were collected from an upper level cardiovascular engineering course at a medium-sized research university. Students learned about the main concepts of cardiovascular engineering including functional block diagrams of the heart, arterial systems, and ECG. The students were assessed on their ability to apply these concepts to a project they worked on during the semester. After identifying a project and a team, they wrote learning objectives and corresponding deliverables that would share what they needed to learn, when they would learn it, and how they would demonstrate it. Students could adjust their learning goals as needed, but they were expected to share their progress on their project multiple times during the semester [14]. To get an A in the course, students needed to work on an innovation project and achieve high external value. High external value is demonstrated by sharing your work outside the course and getting review from a subject-matter expert, e.g. publishing a paper, presenting at a conference, or submitting an invention disclosure [2,19].

3.2 Online Learning Portal

A custom online learning management system (LMS) was created for students to keep track of their learning objectives and deliverables [18]. When students add a learning objective, they give it a title, description, assign it to a level of Bloom's Revised 3D Taxonomy [7], and categorize it using the list of objective categories in Appendix A. When adding a deliverable, students give it a title, description, level of external value, estimated completion time, and status (not started, in progress, or completed). An example of a learning objective and corresponding deliverables is shown in Figure 1. Every time a student adds, edits, or deletes an objective or deliverable, the action is recorded as a log entry, allowing us to see not only the completed products, but also early iterations of the students' objectives [16].

3.3 Data Set

28 students agreed to share their data during the semester. The average student logged approximately 8 objectives, 32 deliverables, and visited the platform more than 65 times during the semester. 17 students achieved high external

Design the Hardware for an ECG Biometric	ES5	Create	Conceptual	Add	View
Choose Analog to Digital Converter	Other	Low	Completed		
Verify ECG Schematic	Other	Low	Completed		
Test PCB	Other	Low	Completed		
Implement the hardware in a working device	Other	Med	In progress		

Figure 1: Example of collected learning objective and corresponding deliverables.

value, 10 students worked on an innovation project but did not achieve high external value, and 1 student made some learning goals but did not complete any.

3.4 Feature Collection

Two main types of features were used and compared: quantitative data and text data. The quantitative features that were extracted from the data include countable features (e.g. number of planned learning objectives, number of logins, etc.), quarter-based progress (e.g. number of deliverables completed during quarter 2, number of learning objectives deleted during quarter 4, etc.), presence of the specific learning objectives as seed in Appendix A (e.g. presence of *Invention Disclosure* objective, number of *Fundamentals of Research* objectives, etc.), and the level of learning as defined by Bloom's Revised Taxonomy and the level of external value.

For the text data, all learning objective and deliverable titles and descriptions were extracted for each student. Using the scikit-learn library in Python, all the words that students wrote in their objectives and deliverables were tokenized, counted, and scaled.

4. EXPERIMENTS

4.1 Models and Feature Sets

In order to predict which students would achieve high external value during the course of the semester, three classifier models were tested: Support Vector Machine (SVM), Logistic Regression (LR), and K-Nearest Neighbors (KNN). These three models were chosen because they have some level of interpretability, an important feature in EDM [15]. In order for instructors to use the discovered information, they need to be able to understand where it was derived from. The baseline model was a Majority Class (MC) classifier.

In addition to comparing the models, both the text and quantitative features were compared. For each set, we also compared using all features to using the top K features. K was optimized and set at 24 for text and 15 for quantitative.

4.2 Evaluation Metrics

Each model was evaluated by calculating accuracy, recall, F1 score, and Area Under Receiver Operating Characteristic Curve (AUC). Accuracy is the proportion of correctly

classified students to all students. Recall is the proportion of students that the model identified as not being on track to success to the number of total students that did not achieve high external value during the course. F1 score is a performance metric that takes the harmonic mean of precision and recall. AUC is the area under the Receiver Operating Characteristic (ROC) curve which shows how well the model can differentiate between the two classes. All models were evaluated using ten-fold cross validation.

4.3 Trajectory

In addition to exploring models that were developed by using each student's final learning objectives and deliverables, we were also able to explore how prediction power of the models changed during the course of the semester. Models were created using daily snapshots of all students to see when the model can begin predicting student success.

5. RESULTS

5.1 Comparing Models and Feature Sets

Table 2 shows the accuracy, recall, F1 score, and AUC for each of the models and feature sets explored. These classifiers used all available data during the semester. Almost all models performed better than the MC baseline test. The text features consistently performed better than the quantitative features, and using feature selection usually improved the model as well. The top models are SVM and LR, both using the top 24 text features. In addition to having low performance, the quantitative models are also difficult to assess in real time. The most relevant features of the quantitative models can give us some information, but they are not as helpful when making predictions. Therefore, we'll focus on using the text models moving forward.

Feature Type	Model	Accuracy	Recall	F1	AUC
Baseline	MC	.6	-	-	.5
All Text Features	SVM	.783	.85	.758	.831
	LR	.883	.85	.866	.972
	KNN	.583	.95	.533	.700
Top 24 Text Features	SVM	.917	.85	.9	.937
	LR	.917	.85	.9	.952
	KNN	.783	.85	.767	.832
All Quantitative Features	SVM	.7	.6	.648	.704
	LR	.717	.5	.612	.697
	KNN	.567	.7	.482	.523
Top 14 Quantitative Features	SVM	.667	.5	.563	.851
	LR	.7	.5	.597	.798
	KNN	.667	.9	.615	.65

Table 1: Performance metrics for each of the models using end of semester data

5.2 Exploring Model Trajectory

Because the model performs well at the midpoint in the semester, the next experiment explored at what point in the semester top-performers can be differentiated from lower-performers. All models used the 24 top text features. Figures 2 and 3 show the accuracy and AUC of the models over time, respectively. The SVM and LR models improve as the

semester goes on, with the AUC for the models leveling out at about day 55. Therefore, the midpoint of the semester seems to be an appropriate time to use the model, but using it earlier might give mixed results.

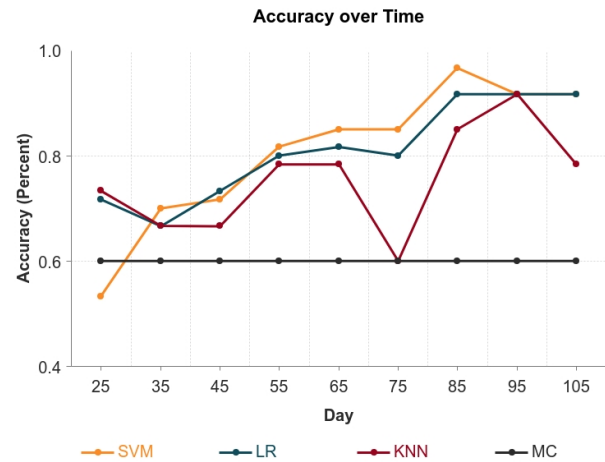


Figure 2: Accuracy of the text-based models over time compared with the baseline MC classifier

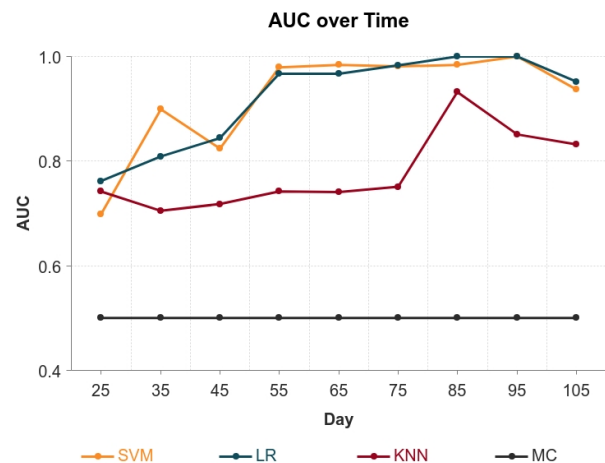


Figure 3: AUC of the text-based models over time compared with the baseline MC classifier

5.3 Knowledge Discovery

In order to better understand what features are most significant in predicting success, the most important features were extracted. By using linear classifier models instead of black-box models like neural networks and other deep-learning models, Chi-Square and the weights of each feature could be calculated. Chi-Square tells us which features are not independent of their classification, meaning they are more likely to differentiate between classes. The greater the Chi-Square value, the greater dependence on classification, meaning that feature is a strong differentiator. Weight can tell us which class a feature is more likely to be found in.

Figure 4 shows the 24 features with the largest Chi-Square value. If the word was more likely to be found in a low-performing student, the Chi-Square value was multiplied by -1 to allow for easier interpretation. The top words that differentiated low-performing students were *information*, *presentation*, *engineering*, *website*, *loops*, *review*, and *feedback*. The top words that differentiated high-performing students were *sensor*, *signal*, *model*, *device*, *idea*, and *symposium*.

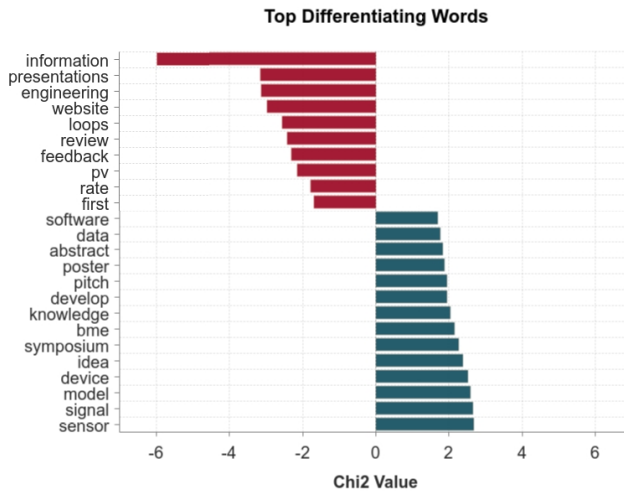


Figure 4: The Top 24 text features that differentiated the most between successful and unsuccessful students. Words with positive Chi-Square values were more associated with successful students. Words with negative Chi-Square values were more associated with unsuccessful students.

Table 4 shows the quantitative features that had the highest Chi-Square values. The weights were used to know which group the variable was more likely to be present in. Top students were more likely to have data analysis, data collection, journal manuscripts, and general Mechanisms of Research learning objectives. Unsuccessful students were more likely to have providing critique and outreach communication learning objectives.

Variable	Chi-Square	Group
Presence of MR4: Data analysis	3.882	Successful
Presence of RM3: Providing critique	3.091	Unsuccessful
Total number of Mechanisms of Research Learning Objectives	2.146	Successful
Presence of MR3: Data collection	1.941	Successful
Presence of PC5: Journal manuscript	1.941	Successful
Presence of PC7: Outreach communication	1.807	Unsuccessful

Table 2: Quantitative features with the highest Chi-Square values

6. DISCUSSION

6.1 Insights Gained

Unsurprisingly, top students were more likely to mention work on their abstracts, posters, pitches, and presence at the BME Symposium (an on-campus biomedical engineering conference). Low-performing students were more likely to have deliverables like websites and outreach activities. Although websites could be high impact deliverables, they can also just be a report of students' lower-level learning. For outreach activities, this can be interpreted broadly and could be outreach to a classmate or small group rather than a visit of high impact. In addition, successful students were more likely to have words related to the design process such as *idea*, *develop*, and *data*. Unsuccessful students were more likely to mention words like *information*, *presentations*, *review*, and *feedback*. We believe these words appeared in low-level students because they were activities required by the class. Therefore, top students did not see the need to write specific learning objectives about them, but lower performing students added them in an attempt to have more items logged.

6.2 Limitations

Just as the world around us is accelerating, so are our students. Therefore, these models will need to continue to evolve and improve as students change their approach to the class. Aiming for consistently high performing models is not a realistic goal for this work. Rather, we can use the knowledge discovery from these models to better understand how students move through these environments and aim to better support them.

6.3 Future Work

In addition to collecting data during more semesters and at more universities, we would also like to explore both clustering and sequential modeling moving forward. By clustering similar students and finding patterns that emerge in successful students in that cluster, we can give personalized feedback that allows students to find success while staying true to their own learning goals.

7. CONCLUSION

Modeling student learning in open-ended learning environments can be challenging, but SVM classifiers show potential in being able to predict which students will be successful in an IBL course. Models had accuracy of over 80% and AUC of over .95 by the midpoint in the semester. This accuracy increased to over 90% by the last few weeks of the semester. By using linear models, we could also gain insight as to what features differentiated between successful and unsuccessful students. Using these results can help instructors know which students could use extra support and lead to more understanding about how students progress through problem-solving environments in general. By understanding how to better support our students in the innovation process, we can foster the next generation of problem-solvers to take on the *Age of Accelerations*.

8. REFERENCES

- [1] ABET. Criteria for accrediting engineering programs.
- [2] E. Alvarez Vazquez, M. Pearson, L. Singelmann, R. Striker, and E. Swartz. Federal funding

- opportunity announcements as a catalyst of students' projects in mooc environments. In *6th International Conference on Learning with MOOCs*. IEEE, 2019.
- [3] I. Arroyo and B. P. Woolf. Inferring learning and attitudes from a bayesian network of log file data. In *AIED*, pages 33–40, 2005.
- [4] F. Bouchet, R. Azevedo, J. S. Kinnebrew, and G. Biswas. Identifying students' characteristic learning behaviors in an intelligent tutoring system fostering self-regulated learning. *International Educational Data Mining Society*, 2012.
- [5] T. L. Friedman. *Thank you for being late: an optimists guide to thriving in the age of accelerations*. Penguin Books, 2017.
- [6] J. S. Kinnebrew, K. M. Loretz, and G. Biswas. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *JEDM| Journal of Educational Data Mining*, 5(1):190–219, 2013.
- [7] D. R. Krathwohl and L. W. Anderson. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Longman, 2009.
- [8] S. M. Land. Cognitive requirements for learning with open-ended learning environments. *Educational Technology Research and Development*, 48(3):61–78, 2000.
- [9] Y. Mao, R. Zhi, F. Khoshnevisan, T. Price, T. Barnes, and M. Chi. One minute is enough: Early prediction of student success and event-level difficulty during a novice programming task. *The 12th International Conference on Educational Data Mining*, pages 119–128, 2019.
- [10] N. A. of Engineering. *The engineer of 2020: Visions of engineering in the new century*. National Academies Press, 2004.
- [11] E. Panadero. A review of self-regulated learning: Six models and four directions for research. *Frontiers in psychology*, 8:422, 2017.
- [12] H. J. Passow and C. H. Passow. What competencies should undergraduate engineering programs emphasize? a systematic review. *Journal of Engineering Education*, 106(3):475–526, 2017.
- [13] P. I. Pavlik Jr. Mining the dynamics of student utility and strategy use during vocabulary learning. *JEDM| Journal of Educational Data Mining*, 5(1):39–71, 2013.
- [14] M. Pearson, E. Swartz, R. Striker, E. Alvarez Vazquez, and L. Singelmann. Driving change using moocs in a blended and online learning environment. In *6th International Conference on Learning with MOOCs*. IEEE, 2019.
- [15] C. Romero and S. Ventura. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(1):12–27, 2013.
- [16] L. Singelmann, E. Swartz, M. Pearson, R. Striker, and E. Alvarez Vazquez. Design and development of a machine learning tool for an innovation-based learning mooc. In *6th International Conference on Learning with MOOCs*. IEEE, 2019.
- [17] D. Spikol, E. Ruffaldi, G. Dabisias, and M. Cukurova. Supervised machine learning in multimodal learning analytics for estimating success in project-based

- learning. *Journal of Computer Assisted Learning*, 34(4):366–377, 2018.
- [18] R. Stiker, M. Pearson, E. Swartz, L. Singelmann, and E. Alvarez Vazquez. 21st century syllabus: Aggregating electronic resources for innovation based learning. In *6th International Conference on Learning with MOOCs*. IEEE, 2019.
- [19] E. Swartz, M. Pearson, R. Striker, L. Singelmann, and E. Alvarez Vazquez. Innovation-based learning on a massive scale. In *6th International Conference on Learning with MOOCs*. IEEE, 2019.
- [20] P. H. Winne and R. S. Baker. The potentials of educational data mining for researching metacognition, motivation and self-regulated learning. *JEDM| Journal of Educational Data Mining*, 5(1):1–8, 2013.

APPENDIX

A. LEARNING OBJECTIVE CATEGORIES

Category	Code	Objective
Discipline-Specific Knowledge	DSK0	Cardiovascular Concepts
	DSK1	Learning in student's program
	DSK2	Learning in student's College
	DSK3	Learning outside of College
	DSK4	Freeform learning
Fundamentals of Research	FR1	Research method
	FR2	Literature review
	FR3	Experimental design
	FR4	Experimental equipment
	FR5	Intellectual merit
	FR6	Broader impact
	FR7	IRB/IACUC
	FR8	Lab safety
Mechanisms of Research	MR1	Statistics
	MR2	Experimental controls
	MR3	Data collection
	MR4	Data analysis
	MR5	Drawing conclusions
	MR6	Knowing nature of results
Professional Communication	PC1	Conference abstract
	PC2	Conference poster
	PC3	Conference presentation
	PC4	Proposal presentation
	PC5	Journal manuscript
	PC6	Standard operating procedure
	PC7	Outreach communication
	PC8	Invention disclosure
Research Mindset	RM1	Personal statement
	RM2	Receiving critique
	RM3	Providing critique
	RM4	Metacognition
	RM5	Establishing requirements
	RM6	Team conduct
	RM7	Mindset
Entrepreneurial Skills	ES1	Business model
	ES2	Customer communication
	ES3	Customer segment
	ES4	Value proposition
	ES5	Product evaluation

Table 3: List of all learning objective categories

Linguistic Changes across Different User Roles in Online Learning Environment. What do they tell us?

Lavendini Sivaneasharajah, Katrina Falkner, Thushari Atapattu

The University of Adelaide

{lavendini.sivaneasharajah, katrina.falkner, thushari.atapattu}@adelaide.edu.au

ABSTRACT

In recent years, we have witnessed an increasing interest in online learning environments, particularly in Massive Open Online Courses (MOOCs). However, prevailing studies show that lower percentage of students complete their courses successfully in online learning environment. The vast amount of student data available in MOOC platforms enables us to gain insight into student learning behaviours. In this paper, we explore the idea of ‘student roles’, identifying linguistic change associated with roles that will later help us to understand students’ learning process in MOOCs. As an initial stage of this research, the study aims to categorise student roles (e.g. information seeker, information giver) using discourse analysis, and to further analyse the linguistic change for each student role with time. A multi-class classifier has been built to identify user roles with 82.20% F-measure. Further, our study on linguistic changes demonstrates that distinctive behaviors can be observed across different user roles. Prominent observations include discourse complexity, lexical diversity, level of information embeddedness and lexical frequency profile being high in information giver in comparison to information seeker and other user roles.

Keywords

MOOCs, Discussion forums, Student Role, Natural Language Processing, Machine Learning

1. INTRODUCTION

With the advent of Massive Open Online Courses (MOOCs) there has been an eruption in learning environment [10]. Students are increasingly seeking alternative learning mediums, with MOOCs increasingly looked upon as a valuable source of learning. As many of the MOOCs are freely available for students, it draws interest of thousands of learners.

According to the statistics, over 101 million learners are globally registered to study using MOOCs by the year of 2018 [16]. However, studies show that only one in every twenty students who enrol in MOOCs complete their studies successfully [9]. The participation in MOOCs seems complex with students’ enrolment for varying purposes and varying

intentions [17]. Completion is not necessarily the only indicator of learning success. Knowing that students may enroll to courses for other purposes, we need to explore other perspectives of learning success beyond completion.

The primary problem aims to solve by this research is whether analysing the student role and its associated linguistic change can be used to understand student learning. We believe learner role can give us an indication on whether learning gain is important to measure learning success. We try to answer “Moving between roles is potentially an indication of learning gain”. This hypothesis has not been explored yet in prevailing literature.

As the studies discuss in this paper demonstrate a proof of concept, our initial stage is to identify user roles and a sample of linguistic indicators that are associated with these roles. Our overarching goal is to track these roles and their associated linguistic changes with time. And eventually, predicting the grades for student using these discourse features. We assume observing these roles and associated linguistic changes will eventually result in a deeper understanding of the student’s learning lifecycle.

Hecking et al. [5] identifies these roles with both linguistic and community-related features (e.g. votes, views). However, in a real time system, it is not realistic to wait for the community-related features to classify students into different roles as structural features can be generated throughout the course and they may change with time. Therefore, we intend to identify student roles in MOOC discussion forums solely based on a discourse analysis.

Few research studies [3; 4] have focused on linguistic changes that occur in online communities. Yet, linguistic changes have not been studied along with students’ role.

With reference to the aforementioned aspects, we aim to answer the following research questions: RQ1: Can we build a model that could predict user roles (information seeking, information giving, other) using linguistic only features? RQ2: Do linguistic indicators change significantly across user roles?

Answering these questions will result in identifying user roles and its associated linguistic changes in discussion forums. These studies may assist to understand students’ learning in online learning environment.

The contribution of this work includes a multi-class classification model that uses linguistic-only features to predict user roles in MOOC discussion forums. Since it uses linguistic-only features, our model can be applied to any online forums (e.g. technical forums) for role prediction. Further, we examine the linguistic indicators and its changing patterns associated with user roles at this stage with the intention of proposing a framework in future to understand students’ learning.

Lavendini Sivaneasharajah, Katrina Falkner and Thushari Atapattu "Linguistic Changes across Different User Roles in MOOCs: What do they tell us?" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 667 - 671

2. RELATED WORK

2.1 User role/ Post classification

Searle's taxonomy [15] has been widely used in literature and proven to be a most successful method in speech act classification. From this point, several classification mechanisms have been evolved based on Searle's taxonomy [1; 7].

Hecking et al. [5] have carried out post classification by integrating the categories that prevail in the existing research studies. The study presents three classes (information seeking, information giving and other). It used content-related features (e.g. phrases – “need help or helps you”) and contextual features (e.g. position in the thread, number of votes) for classification purposes and obtained an average of 70% accuracy.

In a real time system, it is not realistic to wait for the contextual features to predict the given classes as they occur throughout the course and changes with time. Therefore, our study focuses solely on the linguistic aspects over contextual and structural features.

2.2 Linguistic change in online communities

An important facet of linguistic research is to identify the correlation between user lifespan and their language use [3; 13]. Given the rich recent work on linguistic analysis in different online communities [3; 6; 13; 14], research scholars also have attempted linguistic analysis in MOOC. Dowell et al. [4] have conducted a study on MOOC data to identify the conversion in learner's language and discourse characteristic with time. However, the research did not investigate the linguistic changes associated with each user role. To address this gap, we conducted several experiments using different linguistic features to discover discourse complexity, lexical diversity, number of embedded information and lexical frequency profile. Even though preliminary work on linguistic change has been conducted in other online communities, there is a lack of work conducted in MOOCs.

3. METHODOLOGY

3.1 Data set

We extracted a dataset from the AdelaideX¹ ‘Introduction to Project Management’ and ‘Risk Management for Projects’ courses offered in 2016 and 2017 respectively. A total of 9,497 user posts was extracted from 923 different users. We sampled 6000 posts from ‘Project Management’ for this study. We extracted user posts of students who have posted a minimum of six posts during the entire semester. Posts were manually annotated as information seeker (IS), information giver (IG) and other (O) user roles by two independent human evaluators. According to Cohens kappa, the high inter-rater agreement ($k=0.924$) between the two annotators ensures the validity of the human annotation.

User role identification

We adopted machine learning techniques to build a multi-class classifier to predict user roles (IG, IS and O) for a given forum post using discourse features and linguistic features that were extracted using Linguistic Inquiry and Word Count (LIWC)

tool². We extracted multiple features to reflect several facets of the text.

We implemented multi-class classifiers using weka for role identification. All classifiers were tested using 10 Fold Cross-Validation to assess effectiveness.

The imbalanced data were handled using Synthetic Minority Oversampling TEchnique (SMOTE). Here we split the data into 70% for training and validation and 30% for testing. Then, we oversample the minority class on each training fold during cross validation. Then, validated the classifier on the remaining fold.

On the other hand, we also performed further analysis on role classification to explore the potential techniques that can be used to address this problem. We implemented multi-class text classification using Keras, a high-level neural network API.

We used existing pre-trained GloVe word embedding to convert the user posts to 100 dimension vectors. Then, built the model with one input layer, one embedding layer and one Long Short-Term Memory (LSTM) layer with 128 neuros and one output layer with three neurons.

3.2 Linguistic study

In our second study, we conducted several linguistic experiments (e.g. discourse complexity, lexical diversity) to understand the linguistic changes of each user role with time.

3.2.1 Discourse Complexity

According to an existing study by Crossley et al. [2], discourse complexity can be measured by several linguistic indicators. One possible way is using any given reading level measures. Therefore, we used Flesch Kincaid [8], a reading level measure and used this measure to explore discourse complexity with time for each user. We also analysed the association between discourse complexity and student roles.

3.2.2 Lexical Diversity

We calculated lexical diversity to measure the vocabulary usage in the given user posts. Measuring the level of lexical diversity requires to quantify how often different kind of words are used in text. According to the prevailing literature [12], lexical diversity can be measured using different formulas such as type-token ration (TTR), measure of textual lexical diversity (MTLD), vocd-D and many. Due to flaws in the traditional methods, we chose MTLD over other lexical diversity measures as MTLD avoids the adverse effects on text length in measuring the lexical diversity.

3.2.3 Lexical Frequency Profile

We have examined the Lexical Frequency Profile (LFP) associated with each user role to understand how well user has written his discourse. Initially, we extracted n-grams from lecture transcripts. We used CountVectorizer to tokenise the text and built a vocabulary list for lecture transcripts.

We created Lexical Frequency Profile for each user post with respect to the given vocabulary list using spaCy³, an advanced Natural Language Processing API. We created a Phrase Matcher Object and applied the matcher object on each user post to

¹ <https://www.edx.org/school/adelaideX>

² <https://liwc.wpengine.com/>

³ <https://spacy.io/>

extract the keywords. Finally, we examined LFP for user roles and its pattern during a role change.

Since these linguistic indicators are normally distributed, we performed One-Way ANOVA to compare the mean value for each variable's distribution. These linguistic indicators are examined during role changes to understand whether there is a significant difference between user roles.

3.2.4 Information Embeddedness

Information embeddedness is one of the key elements that contributes towards student learning. In our study, information embeddedness can be defined as the number of information that can be extracted from any given discourse. This study attempts to find the level of information embeddedness using clause extraction.

Clause extraction has been used in a previous study [11] to determine the relationship between the clauses per sentence and language development. We develop a novel approach in which clauses are been extracted from parse tree using a rule-based approach.

Initially, a pipeline is being built with Part-Of-Speech (POS) tagging, lemmatisation using Stanford CoreNLP⁴ to get the basic interpretation of a student post. Tree Annotation is used to extract a parse tree for a given sentence. Here, we divided a student's post into multiple sentences and identified the number of clauses embedded in each sentence. Initially, clause-level tags (e.g. SBAR) and word-level coordinating conjunction (e.g. CC) have been extracted from the parse tree. Then, we implemented a rule-based approach to extract the number of clauses.

4. Results

The experiment on role identification addresses information seeking, information giving and other role classification solely based on discourse analysis. We analysed the features extracted from LIWC. Further, we performed feature selection technique known as Recursive Feature Elimination with Cross Validation (RFECV) for feature ranking. We performed the feature ranking on 1200 user posts obtained from Risk Management course. According to the RFECV sixteen optimal number of features have been selected. We retrieved the features with highest ranking and fed these features to the classifier.

We conducted Multivariate Analysis of Variance (MANOVA) to measure the significance between linguistic features and user roles. Table 1 presents the top five variables that exhibit the largest effects size along with multivariate F value (Wilks' λ).

Table 1: MANOVA analysis of language variables

Feature	F	η^2
Words per Sentence	754.853*	0.201
Question Mark	505.057*	0.144
Article	493.305*	0.141
Interrogatives	385.516*	0.114
Personal pronouns	294.884*	0.090

*p<0.001

⁴ <https://stanfordnlp.github.io/CoreNLP/>

We implemented multiclass classifiers with different sets of algorithms using Weka. All these classifiers were tested using 10 Fold Cross-Validation to assess the accuracy. Among these, the Random Forest classification model performed best with 82.20 of F measure (see Table 2). Table 2 reports the accuracy, precision, recall and F-measures for different set of classifiers.

Table 2: Results of classifier performance

Classifiers	Accuracy	Precision	Recall	F1	Cohen's Kappa
Naïve Bayes	71.28	74.40	71.30	71.00	0.5117
Random Forest	82.17	82.30	82.20	82.20	0.6955
Simple Logistic	79.35	79.60	79.40	79.40	0.6473
Logistic	79.43	79.70	79.40	79.50	0.6498
SMO	74.80	76.50	74.80	75.30	0.5770

We also performed, the text classification with Keras. As stated above we used GloVe 100 dimension vector to create the vector space for each user posts. We obtained 88.06 as test accuracy. We halt the model from further training to avoid over fitting.

The experiment on linguistic analysis uses different indicators to address the linguistic change associated with each user role.

According to the reading level measures (discourse complexity), the results indicates that if a particular user role can be seen in consecutive posts the level of complexity increases/decreases with minimum change and when there is a role change (e.g. IS \rightarrow IG or IG \rightarrow IS or O \rightarrow IG) there is a dramatic change in discourse complexity. This trend is observed across our data set.

The mean value of Flesch Kincaid Grade Level measure for user roles are as follows: $\mu = 16.15 \pm 12.86$ (IG), $\mu = 8.77 \pm 7.43$ (IS) and $\mu = 5.30 \pm 6.99$ (O). High Flesch Kincaid score indicates the discourse is difficult to understand. This implies that discourse complexity decreases along with these user role changes whereas the readability of the text becomes easier with these role changes. The results from the One-Way ANOVA test show that there is a significant difference in mean values (discourse complexity) with p-value<0.001 among these user roles.

As stated above, lexical diversity of user posts were obtained via calculating 'Measure of Textual Lexical Diversity'. The mean value of MTLT for user roles are: $\mu = 60.845 \pm 32.380$ (IG), $\mu = 52.18 \pm 39.59$ (IS) and $\mu = 34.46 \pm 46.55$ (O). This indicates that lexical diversity of the user roles are decreasing along these role changes.

The results of Lexical Frequency Profile show that the number of lecture related keywords used in user post changes during a role change. For a given user, the number of keywords used in an information giving post increases – reach an optimal number and decreases with time whereas for an information seeking post it increases/decrease with time. Moreover, information giver uses more keywords from the lecture transcript than information seeker and other.

In information embeddedness factor, we extracted the clauses using a rule-based approach. Once the number of clauses been extracted using clause-level tags and rule-based approach, we compared them with user roles (IG, IS, O). Figure 1 shows the level of information embeddedness in a user posts (number of clause) changes with time for sample of three users.

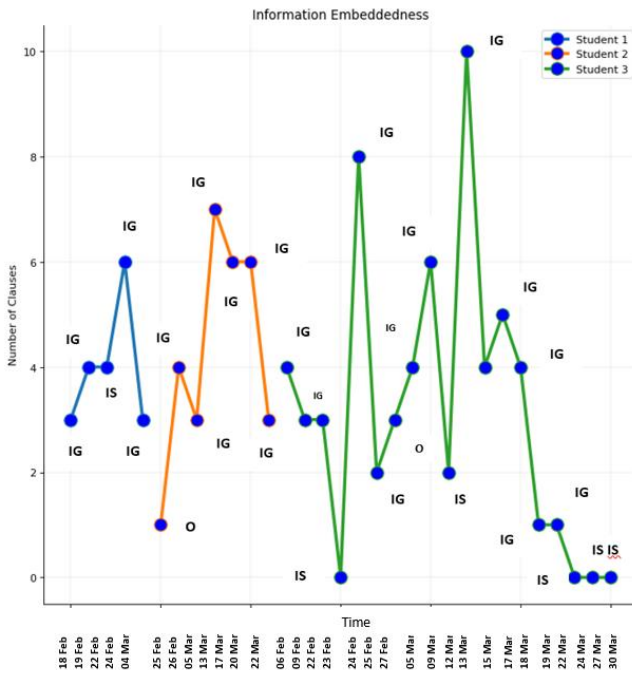


Figure 1: Information embeddedness across user roles with time

5. DISCUSSION

5.1 Can we build a model that could predict user roles using linguistic only features?

In the existing literature [5], user roles have been predicted based on the other contextual features (e.g. votes, views) which delays the predictions in a real time MOOC environment. Therefore, this study extends the line of research to construct a machine learning model to identify user roles (IS/IG/O) using linguistic-only features.

To address RQ1, we conducted an experiment on IS/IG/O post classification. The success of our approach with 82.20% of F-measure and the text analysis performed using Keras with 88.06 accuracy demonstrates that simple linguistic features can be used in role predictions in real time.

According to the feature space used in this classification, it is evident that sixteen identified linguistic features (see Table 1- we presented the first five features that holds largest effects size due to the page limit) can distinguish IG, IS, O posts. For example, information-seeking posts contain high amount of negative emotions comparatively to information giving posts. Likewise, the number of question marks is high in information seeking posts compared to information giving user posts.

5.2 Do linguistic indicators change significantly across user roles?

To investigate RQ2, we conducted several linguistic experiments to explore the linguistic change across user roles.

The results of our linguistic experiments demonstrate that the readability level (μ of Flesch Kincaid Grade Level) of the information giver is high (i.e. discourse complexity is high) when compared to information seeker and other user roles. This implies that there is a high dramatic change in the linguistic complexity during a role change. One possible reason can be information givers tend to include words that are more complex

and provide extensive information when comparing to other user roles.

We further analysed this results by manually analyzing random user posts retrieved from the data set. According to the sample user posts given below, information givers try to elaborate their information with more complex words than information seeker.

Information Giver- "Great use of the likelihood/impact scale! You might also want to use the PESTLE framework to identify broader areas of potential concern..."

Information Seeker- "That was great can i please gain form you, the Challenges you faced during you first project"

Similarly, the results obtained for lexical complexity shows that lexical complexity is higher for information giver. According to the above sample user posts, it is vital that the vocabulary usage is higher in information giver than information seeker.

The trend in the lexical frequency profile shows that information giver uses more keywords from lecture transcripts at the beginning of the course, reaches an optimal point and decreasing afterwards. One possible reason could be that they are enthusiastic to share the lecture related information during the start of the course and it increases with time. Further, the reason to decrease the amount of content-related keywords from the lecture transcript at the end of the course might be because they elaborate concepts in their own words or uses related keywords from other resources as they progress.

We can observe two kind of trends in information seeker. First trend is they use more keywords as they progress. The reason could be, they might not know the content at the beginning but with time, they know the course related keywords. Other trend is they use less keywords with time. The reason might be they try to change their role from the information seeker. Further, we hope to do a meticulous analysis to explore these patterns with the intention of discovering the exact reasons behind them.

In summary, we have achieved the aim of our study as the classifications is purely built upon the idea of utilising linguistic-only features. Further, to understand student learning, we explored RQ2 by examining the different linguistic indicators. These linguistic indicators will have a great potential to understand a user's behavior in any kind of discussion forum.

6. CONCLUSION

We have presented a multi-class user role classification in MOOC discussion forums using linguistic-only features with the intention of eliminating the drawbacks (e.g. contextual features) that exist in previous studies. Our model performed well comparing to base line model with 82.20% of F-measure

On the other hand, our linguistic study gives us a clear differentiation on linguistics aspects associated with each role. The level of information embeddedness, and discourse complexity and lexical diversity of information giver is high compared to information seeker and other. As a proof of concept, our technique demonstrated the potential of identifying the linguistic behaviors for each user role.

This novel approach holds a great promise for user role classification and the associated linguistic behavior in MOOC discussion forums. Additionally, we believe that tracking these role changes and associated linguistic changes will help to understand the student learning in MOOC discussion forums.

7. REFERENCES

- [1] Bhatia, S., Biyani, P., and Mitra, P., 2012. Classifying User Messages For Managing Web Forum Data. In *Proceedings of the Proceedings of the 15th International Workshop on the Web and Databases* (2012), 13–18.
- [2] Crossley, S.A., Greenfield, J., and McNamara, D.S., 2008. Assessing Text Readability Using Cognitively Based Indices. *TESOL Quarterly* 42, 3, 475-493. DOI=<http://dx.doi.org/10.2307/40264479>.
- [3] Danescu-Niculescu-Mizil, C., West, R., Jurafsky, D., Leskovec, J., and Potts, C., 2013. No country for old members: user lifecycle and linguistic change in online communities. In *Proceedings of the Proceedings of the 22nd international conference on World Wide Web* (Rio de Janeiro, Brazil, 2013), ACM, 307-318. DOI=<http://dx.doi.org/10.1145/2488388.2488416>.
- [4] Dowell, N.M.M., Brooks, C., Kovanović, V., Joksimović, S., and Gašević, D., 2017. The Changing Patterns of MOOC Discourse. In *Proceedings of the Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale* (Cambridge, Massachusetts, USA, 2017), ACM, 283-286. DOI=<http://dx.doi.org/10.1145/3051457.3054005>.
- [5] Hecking, T., Chounta, I.-A., and Hoppe, H.U., 2016. Investigating social and semantic user roles in MOOC discussion forums. In *Proceedings of the Proceedings of the Sixth International Conference on Learning Analytics & Knowledge* (Edinburgh, United Kingdom, 2016), ACM, 198-207. DOI=<http://dx.doi.org/10.1145/2883851.2883924>.
- [6] Huffaker, D., Jorgensen, J., Iacobelli, F., Tepper, P., and Cassell, J., 2006. Computational measures for language similarity across time in online communities. In *Proceedings of the Proceedings of the HLT-NAACL 2006 Workshop on Analyzing Conversations in Text and Speech* (New York City, New York, 2006), Association for Computational Linguistics, 15-22.
- [7] Kim, S.N., Wang, L., and Baldwin, T., 2010. Tagging and linking web forum posts. In *Proceedings of the Proceedings of the Fourteenth Conference on Computational Natural Language Learning* (Uppsala, Sweden, 2010), Association for Computational Linguistics, 192-202.
- [8] Klare, G.R., 1974. Assessing Readability. *Reading Research Quarterly* 10, 1, 62-102. DOI=<http://dx.doi.org/10.2307/747086>.
- [9] Koller, D., Ng, A., and Chen, Z., 2013. Retention and Intention in Massive Open Online Courses: In Depth (2013). 'from <https://er.educause.edu/articles/2013/6/retention-and-intention-in-massive-open-online-courses-in-depth>
- [10] Loya, A., Gopal, A., Shukla, I., Jermann, P., and Tormey, R., 2015. Conscientious Behaviour, Flexibility and Learning in Massive Open On-Line Courses. In *Proceedings of the Procedia - Social and Behavioral Sciences* (2015 2015), 519-525. DOI=<http://dx.doi.org/10.1016/j.sbspro.2015.04.686>.
- [11] Lu, X., 2011. A Corpus-Based Evaluation of Syntactic Complexity Measures as Indices of College-Level ESL Writers' Language Development. *TESOL Quarterly* 45, 1, 36-62.
- [12] McCarthy, P.M. and Jarvis, S., 2010. MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods* 42, 2, 381-392.
- [13] Nguyen, D. and Rosé, C.P., 2011. Language use as a reflection of socialization in online communities. In *Proceedings of the Workshop on Languages in Social Media* Association for Computational Linguistics, 76-85.
- [14] Postmes, T., Spears, R., and Lea, M., 2006. The Formation of Group Norms in Computer-Mediated Communication. *Human Communication Research* 26, 3, 341-371. DOI=<http://dx.doi.org/10.1111/j.1468-2958.2000.tb00761.x>.
- [15] Searle, J.R., 1976. A Classification of Illocutionary Acts. *Language in Society* 5, 1, 1-23.
- [16] Shah, D., 2018. By The Numbers: MOOCs in 2018 (2018). 'from <https://www.classcentral.com/report/mooc-stats-2018/>
- [17] Zheng, S., Rosson, M.B., Shih, P.C., and Carroll, J.M., 2015. Understanding student motivation, behaviors and perceptions in MOOCs. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, 1882-1895.

Toward a deep convolutional LSTM for eye gaze spatiotemporal data sequence classification

Komi Sodoké
Université du Québec à
Montréal
2098 Rue Kimberley
Montréal, QC
sodoke.komi@uqam.ca

Aude Dufresne
Université de Montréal
2900 Edouard Montpetit
Montréal, QC
aude.dufresne@umontreal.ca

Roger Nkambou
Université du Québec à
Montréal
2098 Rue Kimberley
Montréal, QC
nkambou.roger@uqam.ca

Issam Tanoubi
Université de Montréal
2900 Edouard Montpetit
Montréal, QC
i.tanoubi@icloud.com

ABSTRACT

Eye gaze movements analysis are being increasingly used in many researches within learning context. Most of those researches analyses the eye movements fixations inside some areas of interest, the saccades trajectory and the scanpath. The eye gaze data are spatiotemporal sequences representing the dynamic of the eye fixations in the visual space over the time. In addition, they contain noises caused by different factors. The task of developing predictive model based on those raw spatiotemporal eye gazes' sequences is challenging. In this research, we present machine learning approaches that we have successfully used to address those challenges with high accuracy mainly with the deep convolutional LSTM architecture.

Keywords

Eye tracking; Deep learning; Spatiotemporal eye gazes sequences classification

1. INTRODUCTION

In some medical field such as anesthesiology, the visual perception is just a tip of the iceberg known as the "situational awareness." In fact, the clinician needs to develop the skills to see adequately the patient vital signs evolution over the time in order to build their understanding and interpretation of the clinical situation to perform their clinical reasoning. In this paper, we explore the following question: Can we tell novice and expert clinicians apart by analyzing only their eye-gaze movements to perform their clinical reasoning? Eye gaze data often contains noise which can be caused by many factors [10]. In addition, the consecutive data points generated by the eye movements trajectory over

the time within the area of interest are spatiotemporal considering their order and their positions in the visual space. Ultimately, our experiments aim to understand key differences between novice and expert clinicians eye movements behavior during their clinical reasoning. Taken together, they will provide us insights to build an Intelligent Tutoring System (ITS) aiming to reinforce gradually the learning curve of novice clinicians with some cues from the experts behavioral implicit knowledge in terms of visual attention to perform a clinical reasoning in critical anesthesiology case.

2. RELATED WORKS

The researches using eye-tracking and ITS can be summaries in two main axes according to Conati et al [6]. The first axe is the investigation of eye-tracking data as source of information for student modelling and personalized instructions. The second axe is leveraging the gaze data to attempt to understand relevant student behaviors. For that purpose, data mining techniques are often used to retrieve similarities, differences, etc. using the eye movements characteristics such as the fixations, the saccades and the scanpaths. Some researches also focus on mining eye-tracking patterns [18]. As a contribution, in this paper we propose predictive models using the sequence of the eye fixations positions over the time. These model will be used by the envisaged ITS to proactively classify eye fixations patterns as Novice vs Expert behavior in order to provide adequate eye movement tutoring services.

3. EXPERIMENTS AND DATASET

3.1 Experiments

An experiment has been conducted to collect eye gaze data for the research using an authentic task involving visual perception and clinical reasoning. Seven Novices and seven experts clinicians were asked to visualize a simulated clinical scenario to perform their clinical reasoning. A [Novice] is a resident clinician within the first or second year of the residency program (PGY1 or PGY2).¹ An [Expert] is a hospital staff member with more than 8 years experience. Each

¹PGY refers to a North American scheme denoting the progress of postgraduates in their residency programs.

Komi Sodoke, Roger Nkambou, Aude Dufresne and Issam Tanoubi "Toward a deep convolutional LSTM for eye gaze spatiotemporal data sequence classification" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 672 - 676

participant looked at a 23" HD monitor (1920x1080 px) on which the simulation was broadcasted. A Tobii TX300 eye tracker was attached to the monitor to record their eye-gaze movements. The simulation is based on the Cannot Intubate/Cannot Oxygenate (CICO) algorithm from the Difficult Airway Society to manage unanticipated difficult intubation in adults [9]. The simulation was scripted to integrate various unanticipated and realistic complications. It was recorded using high-fidelity settings and the video had a total duration of 13 minutes.

As a task, the participants were asked to verbalize their clinical reasoning using a think-aloud protocol (recorded with the eye tracker built-in microphone) while watching the simulation video. Specifically, they had to explain what they see in the different areas of interest (Figure 1) to perform their reasoning. In addition, the participants must explain what they would have done as clinician in charge in some key medical and situational awareness events (Table 1) identified throughout the simulation.

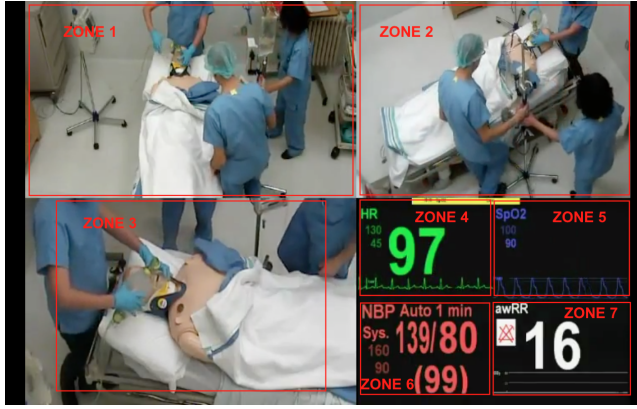


Figure 1: Areas of interest in the simulation

The display screen was divided in seven zones; each representing an area of interest (AOI).

3.2 Dataset

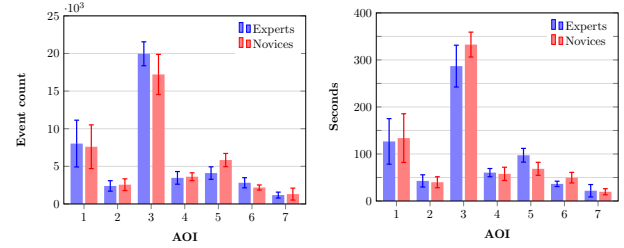
The eye tracker has an accuracy of 0.4deg and was set to a sampling rate of 60Hz. This means that a data point is collect each 17 ms. Each “data point” in the dataset is identified with a $\{x, y, t\}$ tuple by the eye tracker. Overall, our eye-tracking dataset contains about 645k data points; i.e., 14 time series of around 46k points each. Each time series $T = \{p^{(1)} \dots p^{(n)}\}$ is a sequence of 2D vectors, where each vector $p^{(i)} = [x_i, y_i]$ represents the eye-gaze position at a given timestamp t_i .

4. PRELIMINARY ANALYSIS

4.1 Eye movements fixation analysis

First, we conducted preliminary analysis, aimed at providing exploratory insights. For that, we compare novices vs. experts using descriptive statistics on the fixation. For example, the result for the total fixation count and the total fixation duration within each AOI are shown in Figure 2.

These preliminary analysis results showed that both experts and novices have their highest total fixation duration on



(a) Fixation count mean (b) Fixation duration mean

Figure 2: Event count (2a) and mean fixation duration (2b). Error bars denote 95% confidence intervals.

the Technical view (AOI 3) and the General view (AOI 1). This result is further confirmed by the fixation count. Second, novices spent a significantly shorter amount of time at the Saturation view (AOI 5) than the experts ($M = 59$ vs $M = 107$ s, $p = .002$). Inversely, novices spent a significantly higher amount of time at the Technical view (AOI 3) than experts ($M = 382$ vs $M = 266$ s, $p = .042$). All other comparisons were not found to be statistically significant.

4.2 Eye movements behavior around the key events

The video recordings were annotated at different timestamps in terms of clinical keys events. The Table 1 provides an overview of such key event annotations.

Focus Area	AOIs	Time	Description
Healthcare provider	1,3	02:41	Call for help
	1,2,3	03:35	Mask ventilation
	3	06:41	Installation of oropharyngeal cannula
	3	07:35	Use of video-laryngoscope
	1,3	08:33	Use of supra-glottic device
	1,3	09:39	Blue Code initiation
Patient	3	10:32	Initiation of surgical airway
	2	01:10	Impaired verbal response
	3	01:25	Eye closure
Vital signs monitor	1	02:09	Hypoventilation
	5,7	01:37	Desaturation
	4,6	08:33	Bradycardia
	5	10:22	Loss of the saturation signal

Table 1: Key events through the simulation video, together with their relation to the eye tracker AOIs.

With this video annotations, we rendered the heatmaps from raw eye-gaze coordinates corresponding to each key event. We considered eye movement data corresponding to 2 seconds of duration, 1 second before and 1 second after each key event timestamp, given that both eye fixations and reaction times occur typically around 500 ms [8, 15, 19]. Therefore, it allows to capture the eye gaze behaviour before and after each key event.

With this fine-grained video annotations and the observation of the incremental heatmaps around each key events, we observed more salient differences between novices and experts. For instance at 06:41 (Installation of oropharyngeal cannula) we observed a divergent eye movements behavior: both novices and experts focused on AOIs 3 and 5, but novices also focused in AOI 1 (Figure 3). These observations

suggest that both novices and experts have subtle different eye-gaze movement patterns most of the time, while sometimes they are similar. What is most important, these eye-gaze patterns vary over time, suggesting that both novices and experts tend to focus on different AOIs over time.

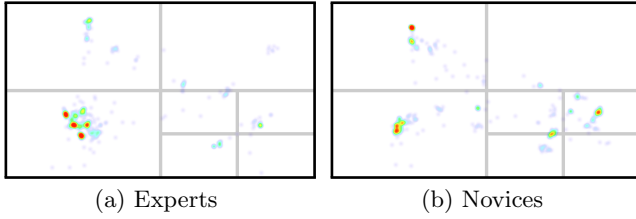


Figure 3: Heatmap of the eye-gaze coordinates taking into account 1 second before and after the key events at 06:41

5. EXPERTISE CLASSIFICATION BASED ON EYE GAZE SEQUENCE COORDINATES

Taken together, the preliminary and the behavioral analysis suggest that we could build a classification model considering the eye-gaze movements coordinates over time. Based on the outcome and observations from the preliminary analysis, we wondered if we could automatically learn these eye-gaze behaviors and discriminate clinicians' expertise accordingly; i.e., given a particular sequence of eye movements with their coordinates, can we predict if it is a novice or an expert eye movements behavior? That research objective is a two-class (binary) classification problem on spatiotemporal eye gaze data.

5.1 The challenges of sequential data classification

As discussed by Xing et al. [26], there are three major challenges in sequence classification. First, the vast majority of classifiers can only take input data as a vector of features. However, there are no *explicit* features in sequence data. Second, even with various features selection methods to transform a sequence into a set of features, the feature selection is far from trivial. The dimensionality of the feature space for the sequence data can be very high and the computation can be costly. Third, besides accurate classification results, in some applications, we may also want to get an "interpretable" classifier. As previously stated, building an interpretable sequence classifier is difficult since there are no explicit *a priori* features.

There are many approaches that have been proposed to address the problem of sequence classification. We will briefly discuss the two main categories: vector-based and model-based classification. In vector-based classification, a data sequence is transformed into a vector of features through feature selections. Then, we need a distance function to measure the similarity between a pair of sequences. The choice of distance measures is critical to the performance of these classifiers. For simple time series classification, Euclidean distance is a widely adopted option [26]. Since Euclidean distance is sensitive to distortions in time dimension, dynamic time warping (DTW) is proposed to overcome this problem and does not require two time series to

be of the same length [13]. Dynamic time warping is usually computed by dynamic programming and has the quadratic time complexity. Therefore, it is computationally costly on a large data set. Using that vector representation of the data, sequences can be classified by a conventional classification method, such as support vector machines [24], decision trees [4], etc.

In model-based classification, given a class of sequences, an underlying model learns the probability distribution of each sequence. The simplest approach is the Naive Bayes sequence classifier [7]. It assumes that, given a class, the features in the sequences are independent of each other. However, this assumption is often violated in practice. A hidden Markov model (HMM) can learn the dependence among elements in sequences [1, 22], assuming that the system being modelled is a Markov process with unobserved states, where the state is described by a single discrete random variable. In contrast, neural networks do not have these assumptions. Moreover, HMMs can only deal with a limited number of step dependencies, while LSTMs can deal with long-term dependencies.

5.2 Machine Learning Models

Since the objective is to predict the expertise given a particular eye movements sequence, the full-length eye-gaze sequence are sliced in smaller parts. Each instance is a fixed-size time series consisting of the raw eye-gaze coordinates; i.e., (x, y) points (a 2D vector). For our experiment, we used sequence slices of length $s = 1000$, which represent eye-gaze sequences (time series) of about 17 seconds each. Finally, because of the small number of participants, we choose the LOOCV (Leave-one-out Cross Validation) as a resampling technique.

Two machine learning architectures were developed to perform the eye gaze spatiotemporal data classification: a WKM-kNN architecture and a DeepConv-LSTM architecture

5.2.1 WKM-kNN architecture

The WKM-kNN architecture is a composition of warped K-means (WKM) with k-nearest neighbor (k-NN). WKM is a fast algorithm for clustering data sequences based on distances, and has outperformed comparable approaches in the task of sequence classification [17]. In addition to providing a compact representation of data sequences, WKM makes them robust to noise or distortions in such data. The input to this model is a time series (a sequence of 2D vectors), and the output is either novice or expert, according to the k-NN classifier.

The WKM algorithm capitalizes in the sequentiality of the data and starts with a suitable initial partition [16], by using piecewise linear interpolation, which results in a non-linearly distributed initial partition of the data. Then, WKM iterates over the data points using a K-means-like optimization procedure. Finally, the k-NN classifier is a non-parametric instance-based learning method, which is among the simplest of all machine learning algorithms. In this work we use $k = 1$ for classification.

To sum up, the WKM-kNN architecture proceeds as follows: first WKM compresses a time series of length n into c disjoint homogeneous segments (or “elementary units”) with $1 < c \ll n$, then the centroid of each segment is used as input to a 1-NN classifier. As in any other clustering algorithm, the number of sequence chunks c should be provided as input. Therefore, because the optimum c for classification is unknown in advance, we tested different values of c , increasingly from 1 (each time series is reduced to a single 2D vector) to 500 (half of the original sequence length).

5.2.2 DeepConv-LSTM architecture

The DeepConv-LSTM architecture is a neural network consisting of a convolutional block followed by a recurrent block (Figure 4).

The recurrent block is a deep long short-term memory (LSTM) network. LSTMs are a type of recurrent neural networks (RNNs) capable of learning long-term dependencies in time series by selectively remembering patterns for long duration and were developed to deal with the *exploding* and *vanishing gradient* problems of traditional RNNs [2, 20]. LSTMs have outperformed many other approaches in a variety of tasks, such as handwriting [11] and speech recognition [12], therefore we adopted this model to analyze eye-gaze sequences. In addition, inspired by recent work that has applied convolutional neural networks (CNNs) to sequence modeling with great success [3], we add a one-dimensional convolutional layer (temporal convolution) to the network input followed by a max pooling layer, which then feed the consolidated features to the LSTM. In other words, a CNN layer learns spatial features which are then learned as sequences by an LSTM layer. This way, we combine the spatial structure learning properties of CNNs with the sequence learning of LSTMs. On the other hand, the max pooling layer is a sample-based discretization process, with 3 goals in mind: (1) reduce the input dimensionality, by filtering the initial data representation; (2) avoid over-fitting, by providing an abstracted form of the data representation; and (3) lower the computational cost, by reducing the number of parameters to learn.

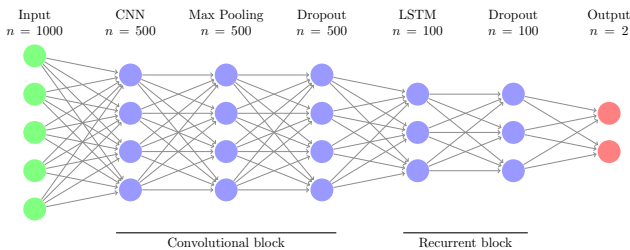


Figure 4: Deep learning network topology. Notes: The drawing is simplified to avoid visual clutter. Each layer dimensionality (n) is denoted below their title.

Overall, the chosen network has 41311 trainable parameters with the topology shown in Figure 4. The network input is the sequence slices (a sequence of 2D vectors), whereas the network output is either novice or expert. Both the CNN and max pooling layers have a kernel size of 2. The LSTM layer is fully connected with 100 neurons. The dropout layers have a probability of 0.2, since it is the recommended

value for most machine learning scenarios; see e.g. [21, 23]. These layers have the effect of reducing overfitting and improving model performance.

We trained the neural network with 60 epochs and a batch size of 256 (mini-batch training) on an i5 CPU @ 3.30 GHz with 16 GB of RAM. After each epoch, the model is evaluated against the testing partition, to get an idea of how well the model is performing during training, after which the data is shuffled for the next epoch. The model was fit using the efficient ADAM optimization algorithm [14] with binary crossentropy as loss function.

5.3 Results

The Table 2 summarizes the results, in terms of classification accuracy. Together with the confidence intervals, we report the Area Under the ROC Curve (AUC), which is a one of the standardized measure of a classifier’s performance. Since the WKM-kNN architecture was tested at different segmentation values c , we report the best classification accuracy result, which was achieved with $c = 4$ segments.

Model	Accuracy (%)	95% Conf. Int.	AUC
WKM-kNN	72.6	[71.1, 74.2]	0.74
DeepConv-LSTM	84.2	[84.9, 86.4]	0.86

Table 2: Summary of the classification results. Confidence intervals are calculated according to the Wilson method for binomial distributions [25].

6. CONCLUSION AND FUTURE WORKS

This research objective is to collect factual eye gaze data from clinicians during a clinical reasoning task. Given a particular sequence of eye movements, with their coordinates; can we predict if it is a novice or an expert clinician eye movements behavior? To answer that question, we built two machine learning models for the binary classification. The deep learning architecture provides an overall better results achieving a very competitive level of accuracy (84.2%) on eye-gaze spatiotemporal data. These results are particularly striking given the fact that we used the *raw* gaze coordinates coming from the eye tracker. The key for the success of a deep neural network classifier is the ability to automatically learn hidden features or intermediates representations in the input data.

The future work is to use the eye-gaze spatiotemporal data classifier outcome and the recorded expert clinical reasoning during the key events as one of the key milestone for the ITS domain model. Also, we have not studied the impact that eye-gaze sequence length may have on model accuracy, though in general shorter sequences should be harder to classify. Some studies argue that humans make informed decisions in a matter of milliseconds [5] although we suspect this is strongly correlated to the application at hand. Therefore, analyzing this possible impact of sequence length on accuracy is another interesting avenue for future work, which in turn opens many research questions. For example: What is the minimum sequence length that maximizes classification accuracy? Is there any upper bound from which we can devise useful eye-gaze information? Does more segment context overlap lead to better model generalization?

7. REFERENCES

- [1] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *Ann. Math. Stat.*, 37(6):1554–1563, 1966.
- [2] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.*, 5(2), 1994.
- [3] J. Bradbury, S. Merity, C. Xiong, and R. Socher. Quasi-recurrent neural networks. *CoRR*, 1611.01576, 2016.
- [4] N. A. Chuzhanova, A. J. Jones, and S. Margetts. Feature selection for genetic sequence classification. *Bioinformatics*, 14(2):139–43, 1998.
- [5] M. Cohen, E. C.E., and F. J. Oscillatory activity and phase-amplitude coupling in the human medial frontal cortex during decision making. *J. Cogn. Neurosci.*, 21(2):390–402, 2009.
- [6] C. Conati, N. Jaques, and M. Muir. Understanding attention to adaptive hints in educational games: An eye-tracking study. *International Journal of Artificial Intelligence in Education*, 23:136–161, 2013.
- [7] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian”on the optimality of the simple bayesian classifier under zero-one loss” classifier under zero-one loss. *Machine Learning*, 29(2–3):103–130, 1997.
- [8] A. H. Duc, P. Bays, and M. Husain. Eye movements as a probe of attention. *Prog. Brain Res.*, 171:403–11, 2008.
- [9] C. Frerk, V. Mitchell, A. McNarry, C. Mendonca, R. Bhagrath, A. Patel, E. O’Sullivan, N. Woodall, and I. Ahmad. Difficult airway society 2015 guidelines for management of unanticipated difficult intubation in adults. *Br. J. Anaesth.*, 115(6):827–48, 2015.
- [10] J. Goldberg and J. Helfman. Comparing information graphics: A critical look at eye tracking. *Conference on Human Factors in Computing Systems - Proceedings*, 04 2010.
- [11] A. Graves, S. Fernández, M. Liwicki, H. Bunke, and J. Schmidhuber. Unconstrained online handwriting recognition with recurrent neural networks. In *Proc. Intl. Conf. on Neural Information Processing Systems, NIPS’07*, pages 577–584. Curran Associates Inc., 2007.
- [12] A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, ICASSP’13*, pages 6645–6649. IEEE Press, 2013.
- [13] E. J. Keogh and M. J. Pazzani. Scaling up dynamic time warping for datamining applications. In *Proc. Intl. Conf. on Knowledge Discovery and Data Mining, KDD’00*, pages 285–289. ACM Press, 2000.
- [14] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *Intl. Conf. on Learning Representations, ICLR’15*. arXiv, 2015.
- [15] R. J. Krauzlis, L. Goffart, and Z. M. Hafed. Neuronal control of fixation and fixational eye movements. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.*, 372(1718), 2017.
- [16] M. H. Kuhn, H. Tomaschewski, and H. Ney. Fast nonlinear time alignment for isolated word recognition. In *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, ICASSP’81*, pages 736–740, 1981.
- [17] L. A. Leiva and E. Vidal. Warped k-means: An algorithm to cluster sequentially-distributed data. *Inf. Sci.*, 237(10):196–210, 2013.
- [18] A. Li, Y. Zhang, and Z. Chen. Scanpath mining of eye movement trajectories for visual attention analysis. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 535–540, 2017.
- [19] J. L. Orquin and S. M. Loose. Attention and choice: A review on eye movements in decision making. *Acta Psychol.*, 144:190–206, 2013.
- [20] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *Proc. Intl. Conf. on Machine Learning, ICML’13*, pages 1310–1318. JMLR.org, 2013.
- [21] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour. Dropout improves recurrent neural networks for handwriting recognition. *CoRR*, 1312.4569, 2013.
- [22] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.
- [23] N. Srivastava. *Improving neural networks with dropout*. PhD thesis, University of Toronto, 2013.
- [24] V. Vapnik. *The nature of statistical learning theory*. Springer, 1995.
- [25] E. B. Wilson. Probable inference, the law of succession, and statistical inference. *J. Am. Stat. Assoc.*, 22:209–212, 1927.
- [26] Z. Xing, J. Pei, and E. Keogh. A brief survey on sequence classification. *SIGKDD Explor. Newsl.*, 12(1):40–48, 2010.

qDKT: Question-Centric Deep Knowledge Tracing

Shashank Sonkar¹, Andrew E. Waters^{1,2},

Andrew S. Lan³, Phillip J. Grimaldi^{1,2}, Richard G. Baraniuk^{1,2}

¹Rice University, ²OpenStax, ³University of Massachusetts Amherst
ss164@rice.edu, aew2@rice.edu, andrewlan@cs.umass.edu, pjg3@rice.edu, richb@rice.edu

ABSTRACT

Knowledge tracing (KT) models, e.g., the deep knowledge tracing (DKT) model, track an individual learner's acquisition of skills over time by examining the learner's performance on questions related to those skills. A practical limitation in most existing KT models is that all questions nested under a particular skill are treated as equivalent observations of a learner's ability, which is an inaccurate assumption in real-world educational scenarios. To overcome this limitation we introduce *qDKT*, a variant of DKT that models every learner's success probability on individual questions over time. *qDKT* incorporates graph Laplacian regularization to smooth predictions under each skill, which is particularly useful when the number of questions in the dataset is big. *qDKT* also uses an initialization scheme inspired by the fastText algorithm, which has found great success in a variety of language modeling tasks. Our experiments on several real-world datasets show that *qDKT* achieves state-of-art performance predicting learner outcomes. Thus, *qDKT* can serve as a simple, yet tough-to-beat, baseline for new question-centric KT models.

1. INTRODUCTION

Knowledge tracing (KT) models are useful tools which provide educators with actionable insights into learners' progress [21, 16]. Given a learner's performance history, these methods predict their proficiency across a predetermined set of skills (i.e., knowledge components or concepts). One of the most popular methods for tracking this cognitive development is the Bayesian Knowledge Tracing (BKT) framework [3, 15, 24] which applies hidden Markov models [1] to learn each learner's *guess*, *slip*, and *learn* probabilities for each skill. Another approach to modeling the dynamics of skill acquisition is SPARFA-Trace [11] which uses Kalman filtering [9] to model learner skill acquisition. An advantage of SPARFA-Trace is that, unlike BKT models, it can relate individual questions to multiple skills. Recently, deep learning techniques have been applied to the KT problem to create Deep Knowledge Tracking (DKT) [18] which mod-

els the sequence prediction task using a Long Short-Term Memory (LSTM) network [8].

All of the aforementioned KT models track an individual learner's knowledge at the *skill* level. Under the KT framework, the time series data modeled consists of learner skill interaction sequences, given by $X_i = \{(s_t^i, a_t^i)\}_{t=1}^T$ where s_t^i is the skill index attempted by the i^{th} learner at discrete time step t , while $a_t^i \in \{0, 1\}$ is the assessment of the learner's response, with 0 indicating an incorrect response and 1 indicating a correct response.

The key assumption underpinning all of the above models is that all questions nested under a particular skill are equivalent. This assumption, however, is generally unrealistic in real-world educational datasets. First, a mapping of questions to skills is not always available and obtaining such a mapping requires the intervention of subject matter experts, which is both costly and time-consuming. Second, questions in real-world educational datasets are never homogeneous, but rather exhibit significant variations in difficulty and discrimination [5]. In other words, different questions convey differing levels of information about a particular learner's mastery of the underlying skill, and methods for modeling learner's acquisition of skills over time should take such information into account.

However, simply substituting questions for skills in a traditional KT model is insufficient to accomplish the goal of tracking an individual learner's knowledge at the question level. To illustrate this, we selected two commonly used educational datasets, ASSISTments2009 and ASSISTments2017.¹ We first ran the standard DKT model using the skill-level information provided with each dataset. We then re-ran the DKT model but used the question identifiers themselves, rather than the skills, for modeling performance. Concretely, the time series data modeled consisted of learners' question interaction sequences, given by $X_i = \{(q_t^i, a_t^i)\}_{t=1}^T$, where q_t^i denotes the question answered by learner i at time t . The AUC for both of these model variants are shown in Table 1. We note that for the ASSISTments 2017 dataset that this question-centric approach provides a moderate improvement in AUC but for the ASSISTments 2009 dataset the question-centric approach significantly hurt AUC.

To understand why this behavior occurs, we note that the

¹<https://sites.google.com/site/assistmentsdata/home>

Shashank Sonkar, Andrew Lan, Andrew Waters, Phillip Grimaldi and Richard Baraniuk "qDKT: Question-centric Deep Knowledge Tracing" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 677 - 681

Dataset	Number of questions	Avg. Obs. per question	DKT (skill)	DKT (question)
ASSISTments 2017	1,183	145.76	0.72	0.74
ASSISTments 2009	16,891	19.27	0.74	0.68

Table 1: AUC scores for DKT vs. its variant with questions as indices. Using questions indices leads to overfitting when the number of observations per question is small.

average number of observations per question for the ASSISTments 2009 dataset is significantly smaller than that for the ASSISTments 2017 dataset. This results in the question-centric modeling overfitting to the data, which adversely affects predictive accuracy. In contrast, the ASSISTments 2017 dataset has a larger number of observations per question, which helps the question-centric DKT model to avoid overfitting.

It is apparent that question-level modeling has the potential to significantly improve predictive accuracy in KT models as compared to skill-level modeling. However, simply substituting questions for skills in a KT model is insufficient to realize the gain. Addressing this challenge is the focus of our work.

Our main contributions are summarized as follows:

1. We propose a novel algorithm for question-level knowledge tracing, which we dub *qDKT*, that achieves state-of-the-art performance compared to traditional KT methods on a number of real-world datasets.
2. Our method utilizes a novel graph Laplacian regularizer for incorporating question similarity information into *qDKT*. Question similarity can be calculated using the skill information or using textual similarity measures if the dataset contains the actual text for each question. Unlike other KT methods, our method does not assume that each question must be associated with exactly one skill.
3. We propose a novel initialization scheme for question-level KT models using fastText [2], an algorithm for natural language processing (NLP). This initialization scheme learns embeddings that summarize pointwise mutual information statistics [12], which is beneficial for bootstrapping sequence prediction models.

Incorporating question-information to improve skill-centric KT models have been tried in the past, for example, the model proposed by [22] concatenates the question embedding to the skill embedding, which is then used as the input to the model. As training progresses, the model learns both the question embedding, and the skill embedding. However, the focus of our proposed initialization scheme is to bootstrap question-centric KT models without using any skill information. As stated earlier, this is advantageous because firstly, tagging questions with skills can be expensive, and secondly, the design of current skill-centric KT models does not transfer well to question-centric KT models (as shown in Table 1).

Initialized with the fastText-inspired scheme, *qDKT* performs at par with the state-of-art skill-level DKT model on ASSISTments 2009 dataset, and improves it by 5% and 6% on the ASSISTments 2017 dataset and Statics 2011 dataset respectively. Coupling the fastText-inspired scheme with the Laplacian regularizer, *qDKT* gives gains of 2% in AUC score as compared to the skill-centric DKT model for ASSISTments 2009, while also capturing question-specific characteristics.

2. PROBLEM STATEMENT AND DKT OVERVIEW

Each learner’s performance record contains the questions attempted, time at which each question was attempted, and the assessment of each response (either correct or incorrect). Also, assume that the skill associated with every question is known. Given performance records for several learners, one wishes to train a knowledge tracing model with the objective of predicting the success probabilities across the questions (or the skills) at time T for a new learner whose performance history has been recorded until time $T - 1$.

2.1 DKT Model

DKT uses an LSTM to predict a learner’s future performance using their previous assessment history. As discussed earlier, the input to the model is a time series which consists of learners’ skill interaction sequences, given by $X_i = \{(s_t^i, a_t^i)\}_{t=1}^T$. Here we restrict our discussion to a single learner and will omit the superscript i throughout. The forward equations of the DKT model are given by

$$\mathbf{x}_t = W_{xv} \mathbf{v}_t, \quad (1)$$

$$\mathbf{h}_t = \text{LSTM}(\mathbf{x}_t), \quad (2)$$

$$\mathbf{y}_t = \sigma(W_{yh} \mathbf{h}_t + \mathbf{b}_y), \quad (3)$$

where σ is the sigmoid function. In words, the input at time step t is the skill interaction tuple (s_t, a_t) which is encoded by an arbitrary high-dimensional one-hot vector, $\mathbf{v}_t \in \{0, 1\}^{2M}$, where M is the number of skills. Using an embedding matrix, $W_{xv} \in R^{K \times 2M}$, \mathbf{v}_t is mapped to a low-dimensional vector, $\mathbf{x}_t \in R^K$, $K \ll M$ (1), which serves as the input to the LSTM cell. \mathbf{x}_t is passed through each of the input, forget, and output gates and, in the end, the LSTM returns \mathbf{h}_t – the estimate of the learner’s current knowledge state. The final output of the model is $\mathbf{y}_t \in R^M$ which predicts the learner’s success probabilities for all the M skills for the next time step $t + 1$.

2.1.1 Loss in the DKT Model

The output \mathbf{y}_t of the DKT model predicts the learner’s proficiency over the skills for the next time step $t + 1$. During training, the assessment (a_{t+1}) of the learner’s response to the question indexed by q_{t+1} is known beforehand. The success probability for the skill associated with q_{t+1} is given by $y_t[s_{t+1}]$. Since DKT assumes that mastery in the skill is equivalent to mastery in any of the questions under it (i.e., all questions under a skill are equivalent), a trained DKT model should predict the success probability at the skill to be the same as the assessment. This rationale motivates the basis for calculating the loss, ℓ_t , at time t , given by

$$\ell_t = l(y_t[s_{t+1}], a_{t+1}), \quad (4)$$

where ℓ is the binary cross-entropy loss.

2.2 Proposed Model: qDKT

We now introduce our proposed method for KT modeling at the question-level, which we dub qDKT. Our method considers a modified problem statement where we estimate a learner's success probability for each question rather than for each skill. Let a learner's question interaction sequence $X = \{(q_t, a_t)\}_{t=1}^{T-1}$ until time step $T - 1$ be given, where q_t denotes the question answered at time t and $a_t \in \{0, 1\}$ is the assessment of the response to question q_t . Our goal is to output $\mathbf{y}_t \in R^N$ which predicts the learner's success probabilities for all the N questions at the next time step $t + 1$. qDKT utilizes the same architecture as DKT as specified in (1) - (3), but with $\mathbf{v}_t \in \{0, 1\}^{2N}$, $W_{xv} \in R^{K \times 2N}$, and $\mathbf{y} \in R^N$. The updated loss ℓ_t from (4) at time t is then given by

$$\ell_t = l(y_t[q_{t+1}], a_{t+1}). \quad (5)$$

We will refer to this model as the *base qDKT model*, where the prefix q denotes question-level modeling.

3. REGULARIZATION FOR qDKT

As seen in Table 1, the base qDKT model performs poorly for datasets with both a large number of questions and a small number of observations per question. To overcome this, we propose a regularization method for qDKT to combat overfitting. It is reasonable to assume that success probabilities of multiple questions associated with the same skill should not be significantly different for a given learner. Based on this premise, we regularize the variance in success probabilities for questions that fall under the same skill

$$R(\mathbf{y}) = \sum_{i \in Q} \sum_{j \in Q} \mathbf{1}(i, j) \cdot (y_i - y_j)^2, \quad (6)$$

where vector $\mathbf{y} \in R^N$ contains success probabilities of all questions Q in the dataset, $i, j \in Q$ and $\mathbf{1}(i, j)$ is 1 if i, j fall under the same skill, otherwise it is 0.

We add this penalty to the loss and use λ to control the weight of the penalty. Thus, the updated loss function from (4) with the regularization penalty is

$$\ell = l + \lambda \cdot R(\mathbf{y}). \quad (7)$$

3.1 Interpretation of the regularizer

Graph theory provides a clean interpretation for the regularization penalty which is also helpful for speeding up its computation. We construct a graph G with number of nodes equal to the number of questions in the dataset. Two nodes are connected with an edge of weight 1 if the questions are associated with the same skill and with an edge weight of 0 otherwise.

The degree matrix D of a graph G is a diagonal matrix with

$$d_{ii} = \sum_{j \in C_i} w_{ij},$$

where w_{ij} is the similarity between node i and node j (edge weight), C is the set containing all the indices directly connected with i (immediate siblings). The adjacency matrix A

of a graph G stores the edge weights w_{ij} . Given the degree matrix D and the adjacency matrix A of a graph G , the Laplacian matrix L is defined as

$$L = D - A.$$

Then for any vector \mathbf{v} [7],

$$\mathbf{v}^T L \mathbf{v} = \sum_{i,j} w_{ij} \cdot (v_i - v_j)^2. \quad (8)$$

We can then use (8) to simplify the regularization penalty of (6)

$$R(\mathbf{y}) = \sum_{i \in Q} \sum_{j \in Q} \mathbf{1}(i, j) \cdot (y_i - y_j)^2 = \mathbf{y}^T L \mathbf{y}. \quad (9)$$

The simplification of the double summation term to a condensed vector-matrix multiplication term is useful to speed up its calculation, especially while training the qDKT model on GPUs.

Further, our approach to model similarity works even when questions are associated with multiple skills. This provides additional flexibility over previous KT models that restrict each question to be associated to exactly one skill. Such flexibility is important for real-world applications where questions commonly evaluate learners on multiple skills simultaneously. Moreover, this formulation can be helpful to incorporate even other measures of similarity like tf-idf similarity [13] using question text.

4. INITIALIZATION OF qDKT

DKT maps each skill interaction tuple to $\mathbf{x} \in R^d$ via the matrix W_{xv} (see (1)). In DKT, the entries of W_{xv} are initialized with draws from a standard normal distribution. While this approach is straightforward, random embeddings tend to perform extremely poorly in high dimensions where the optimization problem will have an extremely large number of saddle points [4]. To overcome this limitation, we propose a more effective method for initializing W_{xv} inspired by the fastText architecture.

4.1 Language Modeling and fastText

In NLP, language models are used to predict the most likely words that can follow a given sequence of words. Such models are often initialized with word embeddings from algorithms like word2vec [14], fastText and GloVe [17]. At a high level, these algorithms embed words into a high dimensional space such that words that have close semantic relationships will be embedded near one another, while words with low semantic similarity will be embedded further apart [6].

A novelty of fastText is that it considers individual characters in a word when computing the final embeddings. By doing this, fastText recognizes that the words "love", "loved", "lovely", and "lovable" are all related and embed them accordingly.

4.2 Embedding Educational Response Data

In our application, we wish to have a notion of question similarity that can serve to guide our initialization scheme, similar to the notion of similar word contexts in fastText.

Dataset	Learners	Questions	Skills	Records
ASSISTments 2009	4,151	16,891	111	325,637
ASSISTments 2017	1,709	1,183	86	249,105
Statics2011	333	1,223	85	189,297
Tutor	895	5981	1,592	437,524

Table 2: Dataset summary statistics.

To do this, we assemble an approximate “text corpus” from our response data, as follows.

Let set Q contain all the question ids and set U contain all characters. We define a one-to-one mapping $f : Q \rightarrow U$ which maps a question id to a unique character. To convert learners’ question interaction sequences, $X = \{(q_t, a_t)\}_{t=1}^T$ into a text corpus, we apply a signal transformation Y on X such that $y_t = f(q_t) + a_t$ where ‘+’ denotes the string concatenation operator. Thus, each question interaction is encoded as a two character string consisting of the question id and the graded response. This interaction encoding constitutes the “words” of our corpus. The “sentences” of our corpus constitute of the string of such encoded interactions by an individual learner. We finally apply fastText to this newly generated “corpus”. For a given question interaction say $(q, 0)$, fastText will train the embeddings of the following n -grams $\{f(q), '0', f(q) + '0'\}$. Thus, we link the embeddings of $(q, 0)$ and $(q, 1)$ through the embedding of $f(q)$. The resulting output embedding of fastText is used as our initialization of W_{xv} .

5. EXPERIMENTS

5.1 Datasets

We consider four datasets for our experiments: ASSISTments 2009, ASSISTments 2017, Statics 2011, and a dataset from OpenStax Tutor, an online learning platform. The Statics 2011 dataset is from an engineering statics course. Standard pre-processing steps common in the literature are used to clean the data. For ASSISTments2009 dataset, we follow the pre-processing steps recommended by [23]. Duplicated records and scaffolding problems are removed. Also, since the dataset contains a few questions that are associated with multiple skills, those multiple skills were combined into a new joint skill for skill-level DKT models, along the lines of [23]. However, for qDKT, our Laplacian regularization approach provides needed flexibility when questions fall under multiple skills, doing away with the need of combining multiple skill into one joint skill. For the ASSISTments2017 dataset, all scaffolding problems are filtered out. Relevant statistics for each dataset are given in Table 2.

5.2 Experimental Setup and Metrics

Each experiment consists of comparing our proposed qDKT algorithm against the original DKT algorithm for a given dataset. To further quantify the impact of each proposed improvement to the qDKT model we will measure qDKT performance over four different variants: 1) The base qDKT without any regularization and with randomized initialization, 2) qDKT with regularization and randomized initialization, 3) qDKT without regularization but with our proposed initialization scheme and 4) qDKT with both regularization and with our proposed initialization scheme. For all the experiments and datasets, we perform 5-fold cross validation; 70% data is used for training and the rest for

testing. We report the average receiver operating characteristics curve (AUC) score to compare each method. All the models are trained using the Adam optimizer [10] with dropout [20] to reduce overfitting.

5.3 Results and Discussion

Our results are displayed in Table 3. We see that the base qDKT model without regularization and with randomized initialization outperforms the original DKT model on three of the four datasets used. For the ASSISTments 2009 dataset, base qDKT loses by a large margin. This is due to ASSISTments 2009 dataset having a large number of questions coupled with a low number of observations per question (see Table 1). We note that the individual addition of either the regularizer or the fastText initialization scheme greatly improves the performance of qDKT for each dataset. We finally note that the combination of both the regularizer and fastText initialization scheme enables qDKT to achieve better performance than DKT for all datasets considered.

For additional details, please refer to the extended version of this paper [19].

6. CONCLUSIONS

We have proposed qDKT, a novel model for knowledge tracing for educational data. Our method improves on prior art by predicting student performance at the question-level, rather than at the skill level. We have further proposed novel regularization and initialization schemes that greatly improve the performance of our method across several real-world datasets when compared with the traditional knowledge tracing methods. We propose that qDKT can provide a simple, yet tough-to-beat baseline, for new question-centric KT models to come.

Acknowledgements

This work was supported by NSF grants CCF-1911094, IIS-1838177, IIS-1730574, DRL-1631556, IUSE-1842378, NSF-1937134; ONR grants N00014-18-12571 and N00014-17-1-2551; AFOSR grant FA9550-18-1-0478; and a Vannevar Bush Faculty Fellowship, ONR grant N00014-18-1-2047.

7. REFERENCES

- [1] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.
- [2] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
- [3] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.
- [4] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941, 2014.

Dataset	DKT	Base qDKT	Base qDKT w/ Laplacian regularizer	Base qDKT w/ fastText	Base qDKT w/ fastText and regularizer
ASSISTments 2009	0.740 \pm 0.002	0.678 \pm 0.004	0.738 \pm 0.003	0.740 \pm 0.004	0.762 \pm 0.005
ASSISTments 2017	0.721 \pm 0.002	0.742 \pm 0.003	0.753 \pm 0.005	0.772 \pm 0.004	0.770 \pm 0.005
Statics 2011	0.770 \pm 0.003	0.822 \pm 0.003	0.825 \pm 0.002	0.832 \pm 0.003	0.834 \pm 0.002
Tutor	0.856 \pm 0.003	0.875 \pm 0.002	0.882 \pm 0.001	0.890 \pm 0.0008	0.895 \pm 0.001

Table 3: AUC scores for each algorithm and dataset. We see that both the addition of the regularizer and the improved initialization scheme improve performance on all datasets over the original DKT model. Combining both the regularizer and our proposed initialization scheme achieves the best performance over all algorithms.

- [5] S. E. Embretson and S. P. Reise. *Item response theory*. Psychology Press, 2013.
- [6] Y. Goldberg and O. Levy. word2vec explained: deriving mikolov et al.’s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.
- [7] D. J. Hand. Statistical analysis of network data: Methods and models by eric d. kolaczyk. *International Statistical Review*, 78(1):135–135, 2010.
- [8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [9] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.
- [10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015*, 2015.
- [11] A. S. Lan, C. Studer, and R. G. Baraniuk. Time-varying learning and content analytics via sparse factor analysis. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 452–461. ACM, 2014.
- [12] O. Levy and Y. Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.
- [13] J. H. Martin and D. Jurafsky. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson/Prentice Hall Upper Saddle River, 2009.
- [14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [15] Z. Pardos and N. Heffernan. Navigating the parameter space of bayesian knowledge tracing models: Visualizations of the convergence of the expectation maximization algorithm. In *Educational Data Mining 2010*, 2010.
- [16] R. Pelánek. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction*, 27(3-5):313–350, 2017.
- [17] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [18] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in neural information processing systems*, pages 505–513, 2015.
- [19] S. Sonkar, A. E. Waters, A. S. Lan, P. J. Grimaldi, and R. G. Baraniuk. qdkt: Question-centric deep knowledge tracing. *arXiv preprint arXiv:2005.12442*, 2020.
- [20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [21] K. Vanlehn. The behavior of tutoring systems. *International journal of artificial intelligence in education*, 16(3):227–265, 2006.
- [22] T. Wang, F. Ma, and J. Gao. Deep hierarchical knowledge tracing. In *EDM*, 2019.
- [23] X. Xiong, S. Zhao, E. G. Van Inwegen, and J. E. Beck. Going deeper with deep knowledge tracing. *International Educational Data Mining Society*, 2016.
- [24] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *International conference on artificial intelligence in education*, pages 171–180. Springer, 2013.

IntelliMOOC: Intelligent Online Learning Framework for MOOC Platforms

Patara Trirat, Sakonporn Noree, Mun Yong Yi^{*}
Graduate School of Knowledge Service Engineering, KAIST
Daejeon, South Korea
{patara.t, sakonporn.n, munyi}@kaist.ac.kr

ABSTRACT

Massive Open Online Course (MOOC) has been inefficient in responding to students' questions, or in-lesson comments as the volume of questions is truly massive. This paper proposes a framework that utilizes students' behavioral data on the web in addition to text data in answering student questions. With this framework, we built a recommender system that generates a set of ranked video snippets in response to a student's question by implementing a deep neural network for question and confusion classifiers and a content-based recommender for providing answers to the student's question. Preliminary results show that our question and confusion classifiers outperform the baseline models. Our combined recommender model shows the best performance in recommending the answer. As an ongoing endeavor, we are in the process of developing an intelligent agent that leverages the question and confusion classifiers in improving student's achievement.

Keywords

MOOC, Recommender Systems, Question Answering

1. INTRODUCTION

Given the millions of users who are using a Massive Open Online Course (MOOC) platform for their studying, instructors cannot answer all the questions from their students. Consequently, discussion forums are leveraged to facilitate peer-to-peer learning. However, this approach has the potential of misleading each other with inaccurate information as well as the lack of responsibility and participation, thereby contributing to duplicate questions and early dropouts [6]. A few studies developed a question answering model to mitigate the aforementioned problems. YouEDU [1] presented an approach that automatically detects confusion in MOOC forum posts and recommends video clips as answers in a specific course forum. Xiao-Shih [3] is the intelligent educational question answering bot made of Natural Language

^{*}The corresponding author.

Patara Trirat, Sakonporn Noree and Mun Yong Yi "IntelliMOOC: Intelligent Online Learning Framework for MOOC Platforms" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 682 - 685

Processing (NLP) processes and a Random Forest model to answer learners' questions. While these approaches provide some answers to the problem, they primarily targeted at students who actively participate in the course discussion forum. Still, learners who use forums are a very tiny part of course learners. Further, behavioral traces can help identify periods of confusion and the reasons behind [6].

In this paper, to provide a better learning experience on the MOOC platform, we extend prior research by incorporating the idea of detecting student's confusion as in [2, 5, 6, 7] but using web data rather than text data in order to have more responsiveness and interactivity within a single webpage. The methodology and preliminary results are presented in Section 2 and Section 3, respectively.

2. METHODOLOGY

In this section, we present the data set used in this paper, classifiers, recommender models, and ongoing work. We decided to adopt the *Khan Academy* data set, as described in the following subsection. The post data from the discussion forums commonly used by prior work are not employed in our study because our ultimate goal is to develop a single page learning environment for MOOC (Section 2.5) that resembles the Khan Academy environment for a seamless learning experience. Also, given that Khan Academy provides a diverse set of courses in which our approach is validated, using the Khan Academy data set was preferred for generalizability.

2.1 Khan Academy Dataset

As illustrated in Figure 1, we collected the 9,772 videos, 469,474 questions, and 1,048,575 video transcripts through the Khan Academy API¹. Because the length of the given transcripts was too short, only a single sentence for each transcript, we merged them into the list of captions (one-minute long each) by calculating the number of snippets using equation (1).

$$\text{number of snippets} = \left\lceil \frac{t_{\text{end}} - t_{\text{start}}}{60} \right\rceil \quad (1)$$

After that, we used the number of snippets to compute the number of captions by equation (2), resulting in 72,313 captions.

$$\text{number of captions} = \left\lceil \frac{\text{number of transcripts}}{\text{number of snippets}} \right\rceil \quad (2)$$

¹<https://github.com/Khan/khan-api>

Table 1: Description of Features used for training the Confusion Classifier.

Name	Description	Example
replay	Is the video replayed?	0
playback speed	Speed of the video.	0.5
caption	Is caption of the video opened?	1
return	Does a student watch at a previous specific time point?	0
return counts	How many time a student jump to a previous specific time point?	3
forward	Does a student watch at a next specific time point?	1
forward counts	How many time a student jump to a next specific time point?	2
watch counts	How many time a student watch the entire video?	2
pause	Is the video currently paused?	0
pause counts	How many time a student pause the video?	5
volume up	Is the volume increased?	1
volume down	Is the volume decreased?	0
resolution	What is the quality of the video selected?	720

In the question dataset, as many questions were invalid questions (i.e., with the attribute *flags* of inappropriate, comments, misplace, or spam), we utilized a few attributes provided by the Khan Academy API to label each question in building a classifier as follows.

- **flags.** If a user flagged the question, we considered it as a statement. The possible flags are, for instance, *inappropriate*, *changetocomment*, *doesnotbelong*, and *spam*.
- **lowQualityScore.** This attribute shows the quality of the given question. From our observation, we decided to use 0.7 as a threshold, meaning that a sentence with a score of 0.7 or lower is considered a valid question. Further, we noticed that the sentences with the *lowQualityScore* of greater than 10 is also valid. These sentences were all related to a sexual reproduction course and were all valid questions.
- **not_spam.** If value of *not_spam* is true, we considered it as a real question.
- **sum_vote.** The *sum_vote* is incrementally accumulated by the vote of the students (including the one who posts). If the *sum_vote* is greater than 2, we considered it as an actual question.

Finally, we extracted the referenced time from the questions using the regular expression technique when we computed the similarity between the question and captions in the recommendation stage.

2.2 Classifiers

In the classification stage, we set the dependent variable of the data set as a binary class (1 or 0) for both *Question* and *Confusion* classifiers: 1 indicates a real question (by Question Classifier) or student’s confusion (by Confusion Classifier), 0 otherwise.

2.2.1 Question Classifier

We built binary classifiers applying various approaches – both Machine Learning (e.g., Logistic Regression, Random Forest, and SVM) with TF-IDF and Deep Learning (e.g., MLP, CNN, GRU, and LSTM) with the GloVe [4] pre-trained word vector. Regarding training and testing datasets, we had all of the questions go through the NLP processes to extract the tokens of each question. We kept Wh-words and question marks as we found that they had some discriminant power. We used 85% of the dataset as a training set, and the remaining as a testing set.

2.2.2 Confusion Classifier

To build a confusion classifier, we trained bidirectional Gated Recurrent Units (GRU) for classifying the sequences of users’ behavioral log data. As a preliminary evaluation, because the clickstream study data was lacking diverse scenarios, we instead synthesized 100,000 log data (ten sequences each) to simulate the students’ behaviors. The features of the synthesized data are described in Table 1.

2.3 Recommender Models

We built three models, of which the differences were the inputs used to compute the similarity as follows.

- **Baseline.** This model was straightforward. We built it by computing the similarity between the video’s captions and the questions, both of which went through the same NLP steps.
- **Combined.** This model applies the same NLP processes as the previous one, but use more input text. Instead of using only video’s captions, we concatenated the video’s *metadata* to its captions to assign more weights on some specific topics of the video (e.g., Algebra, Renaissance in Italy, and Biology).
- **Noun-based.** This model used the same combination as the previous one but kept only the nouns and noun phrases of the questions and the video’s captions.

In addition to the different input processes of each model, the primary tasks were token vectorization, similarity metric calculation, and time reference extraction. A process after NLP steps was vectorization. We used TF-IDF to build the feature vector of each question and the video’s captions. Subsequently, using the time reference extraction, we concatenated the caption text of the referred time to assign more weights on the specific topic by the specified time in the question. Lastly, we used cosine similarity to calculate the closeness between a question and each of the captions.

2.3.1 Ranking and Recommending Videos

After computing the similarity score between the given question and captions of every video, we sorted the similarity score descendingly in order to select the *top-5 ranked* videos to create a recommendation list. Further, our additional objective was to recommend the videos that can answer the question within a period of one-minute length. Thus, the starting time of the video – that the model ranks and recommends – is the same start time of the caption that matches the question.

2.4 Ongoing Development

We are further working on developing two modules to make the learning environment more interactive and intelligent as follows.

Faster Question Answering. To make the system respond faster and remove potentially duplicated questions more effectively in online settings, we cluster similar questions in meaning that contain the same answer set so that we do not need to compute the similarity between the new question and all of the videos’ captions again. In essence, we

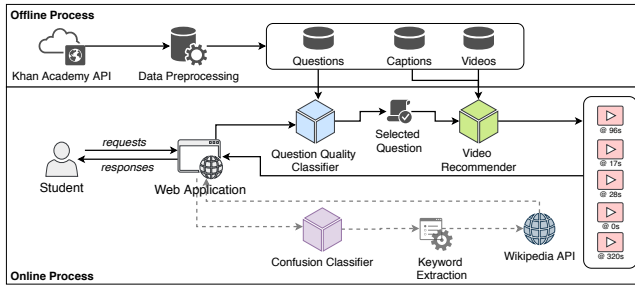


Figure 1: Overview of the proposed framework. The dashed-lines indicate our ongoing work.

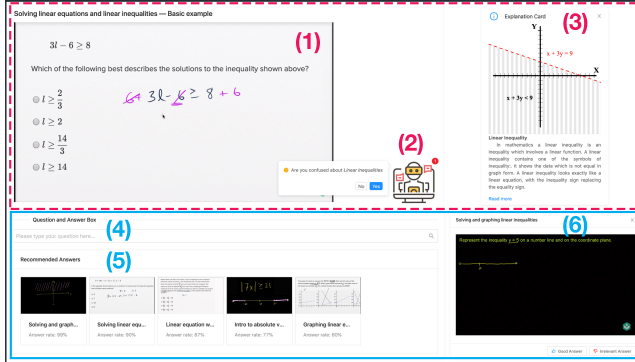


Figure 2: Screenshot of IntelliMOOC prototype.

need to compute only with the question clusters. We also adopt the voting idea to re-compute the weights of the video snippets using the like and dislike interactions as feedback of the student. As a result, students will get the most useful answer at the top while requiring less time.

Intelligent Agent with Confusion Detection. To make it more personalized and interactive, we are in need to develop an intelligent agent, which stays side-by-side to the student, for encouraging the student to ask a question or giving an additional explanation when the student is struggling in a particular topic. For this purpose, we have developed a module that utilizes the feedback from the student’s interaction for re-training the classifier in order to improve the model over time incrementally.

In sum, as shown in Figure 1, we propose a framework based on the techniques above that can work with any MOOC platform by linking the existing discussion forum to the course video pages, so that we can mitigate the dropout and no response problems caused by the confusion that arises during the study [6]. In the next subsection, we describe how we combine those techniques to develop a prototype.

2.5 Prototype of IntelliMOOC

As illustrated in Figure 2, we built a prototype of the IntelliMOOC as the web-based platform consisting of six components in the two modules described above. In the upper segment, it composes of (1) video player, (2) intelligent agent, and (3) explanation card using confusion classifier with keyword extraction and Wikipedia API. In the lower segment, it includes (4) question input box, (5) recommended answer

set, and (6) answered video player using question classifiers with recommendation model. The connection of the underlying processes of each component is depicted in Figure 1. This framework shows how integrating those elements in a single page can provide a better learning experience for the MOOC platform.

3. EXPERIMENTAL EVALUATION

In this section, we show the performance of the classifiers and recommender models. In a standard information retrieval project, the objective is to get the top documents that meet a user’s query. In this work, the query is a question, and the document corresponds to a caption. Our purpose is to retrieve a ranked recommend set of videos that can effectively answer the question.

3.1 Classifiers

We quantified the performance of the classifiers using the two metrics: *Accuracy* and *F1 score*.

Accuracy is the most straightforward standard evaluation metric commonly used for classification models. It measures how the model correctly classified the data.

F1 score is the weighted average score of *Precision* and *Recall* metrics. It is used in this study to examine whether our model still performs well under the class imbalance setting—roughly 3:1 in our case—as it takes both false positives and false negatives into consideration.

Accordingly, we found that the *bidirectional GRU* performed the best in the accuracy and F1 score altogether, achieving **0.84** and **0.78** for the question classifier and **0.997** and **0.99** for the confusion classifier, respectively.

3.2 Recommender Models

We evaluated our recommender models using two metrics: *Parent-Relevancy Score* and *Normalized Discounted Cumulative Gain*. The definition of the two metrics are as follows:

Parent-Relevancy score measures the relevancy between the real topics of the question and the parent topics of the recommended videos. The measurement is divided into two levels. (1) *Root-Level* match is defined as the correct match between the root parent topic of the question and the root parent topic of each video in the recommended set. (2) *1-Level* match is defined as the correct match between at least one parent topic of the question and at least one parent topic of each recommended video regardless of its level.

Normalized Discounted Cumulative Gain (NDCG) computes the sum of the relevance scores (gain) of each recommendation to measure the ranking quality. Nonetheless, the gain is proportionally discounted to how much lower the video is in the ranking. The underlying intuition is that the gain due to a relevant video that appears as an earlier choice should be penalized smaller than it would be if it appeared as a later choice. If $score_i$ is the gain connected with the video at position i , the Discounted Cumulative Gain (DCG) at a position i is defined as:

$$DCG_p = \sum_{i=1}^p \frac{score_i}{\log_2(i+1)} \quad (3)$$

Table 2: Parent-Relevancy scores of each model with the best score obtained by *Combined model*.

	Baseline	Combined	Noun-based
Root-Level	0.667	0.703	0.663
1-Level	0.741	0.760	0.707
Average	0.704	0.732	0.685

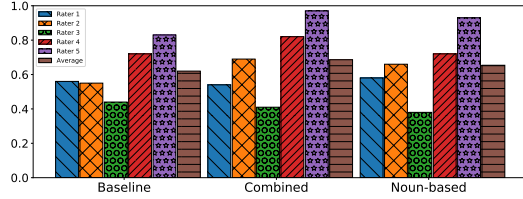


Figure 3: Normalized Discounted Cumulative Gain (NDCG) from each rater with the best score at 0.97 and average score at 0.68, both obtained by the *Combined model*.

We used a score relevance scale of 0, 1, 2, and 3, corresponding to the classes listed below and calculated the DCG for the ranked recommendations we received for each question. The Ideal value of DCG (IDCG) is defined as the DCG based on the ideal ranking as assessed by the raters. To get the IDCG, we order the rankings given by the raters in decreasing order of relevance scores and compute the DCG of the sorted ranking. It corresponds to the maximum theoretically possible DCG in any ranking of the recommendations for the given question. We normalize the DCG for our ranking by the IDCG to make the Normalized DCG (NDCG):

$$NDCG_p = \frac{DCG_p}{IDCG_p} \quad (4)$$

If there are n recommended videos, then we report $NDCG(n)$ as $NDCG$, the overall rating for the ranking.

To evaluate each model, we randomly sampled questions from the Khan Academy data set. Regarding *Parent-Relevancy Score*, we randomly chose 50 questions out of 353,067 questions in the data set and performed five-time iterations to get the average score of each model. In the case of $NDCG$, we randomly selected eight questions from the data set and two new questions from the raters. The raters comprised of one undergraduate and four graduates as the courses were mostly high school courses and introductory undergraduate courses. Each recommender would output the result sets to each rater. They independently evaluated the relevance of each recommended video to the given questions. This process yielded a human-generated ranking, which we then compared to the algorithm's rank order. The rating scale given to the raters is shown below, which is similar to [1]:

- 3: **Completely Relevant** the recommended snippet precisely answer the question.
- 2: **Relevant** the recommended snippet is somewhat useful for answering the question.
- 1: **Somewhat Relevant** the title of the recommended snippet is only relevant to the question.
- 0: **Not Relevant** the recommended snippet is not relevant to the question.

As shown in Table 2 and Figure 3, the *Combined Model* is the best model in recommending ranked video clips in each of the evaluation metrics.

4. CONCLUSION

We propose a framework that includes a recommender model, which answers a student question by recommending a set of relevant video snippets. The experiments showed promising results for both of the question and confusion classifier as well as the recommender model. In particular, the *Combined Model*, which utilizes both of the part of speech (noun, verb, and adjective) and video metadata, produced the best results, outperforming the baseline and noun-based models. Our ongoing research is being carried out to enhance the student learning experience by integrating an intelligent agent into the system, which can timely detect a student's confusion using web data.

5. REFERENCES

- [1] A. Agrawal, J. Venkatraman, S. Leonard, and A. Paepcke. Youedu: Addressing confusion in MOOC discussion forums by recommending instructional video clips. In *Proceedings of the 8th International Conference on Educational Data Mining, EDM 2015, Madrid, Spain, June 26-29, 2015*, pages 297–304, 2015.
- [2] T. Atapattu, K. Falkner, M. Thilakarathne, L. Sivaneasharajah, and R. Jayashanka. An identification of learners' confusion through language and discourse analysis. *arXiv:1903.03286*, 2019.
- [3] H.-H. Hsu and N.-F. Huang. Xiao-shih: The educational intelligent question answering bot on chinese-based moocs. In *2018 17th IEEE Int. Conference on Machine Learning and Applications (ICMLA)*, pages 1316–1321. IEEE, 2018.
- [4] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [5] Z. Shi, Y. Zhang, C. Bian, and W. Lu. Automatic academic confusion recognition in online learning based on facial expressions. In *2019 14th Int. Conference on Computer Science & Education (ICCSE)*, pages 528–532. IEEE, 2019.
- [6] D. Yang, R. E. Kraut, and C. P. Rosé. Exploring the effect of student confusion in massive open online courses. In *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016, Raleigh, North Carolina, USA, June 29 - July 2, 2016*, page 8, 2016.
- [7] Z. Zeng, S. Chaturvedi, and S. Bhat. Learner affect through the looking glass: Characterization and detection of confusion in online courses. In *Proceedings of the 10th International Conference on Educational Data Mining*, 2017.

Using Association Rule Mining to Uncover Rarely Occurring Relationships in Two University Online STEM Courses: A Comparative Analysis

Hannah Valdiviejas

Department of Educational Psychology University of Illinois at Urbana–Champaign, IL, USA
hsv2@illinois.edu

Nigel Bosch

School of Information Sciences and Department of Educational Psychology University of Illinois at Urbana–Champaign, IL, USA
pnb@illinois.edu

ABSTRACT

Metacognition is a valuable tool for learning, particularly in online settings, due to its role in self-regulation. Being metacognitive is especially crucial for students who face exceptional difficulties in academic settings because it grants them the ability to identify gaps in their knowledge and seek help during difficult courses. Here we investigate metacognition for one such group of students: college students traditionally underrepresented in STEM (UR-STEM) in the context of two online university-level STEM courses. Using an automatic detection tool for metacognitive language, we first analyzed text from discussion forums of the two courses; one as a prototype and another as a replication study. We then used association rule mining to uncover fine-grained relationships in the online educational context between underrepresented STEM student status, online behavior, and self-regulated learning. In some cases, we inverted association rules to find associations for underrepresented minoritized students. Implications of the results for teaching and learning STEM content in the online space are discussed. Finally, we discuss the issue of using association rule mining to analyze commonly occurring patterns amongst an uncommon smaller subset of the data (specifically, underrepresented groups of students).

Keywords

Metacognition, Association rule mining, Rare itemsets, STEM

1 INTRODUCTION

The troubling underrepresentation of certain groups of people in STEM majors and careers is a multifaceted and complex issue that does not have one single cause and therefore one single solution. Thus, in this paper we utilize a multi-step research design that involves innovative ways to capture what may or may not be contributing to the underrepresentation of certain students in STEM, specifically through online STEM courses at the university level. In the current study we use student demographic data to understand fine-grained relationships in online learning behaviors, analyzed in ways that are not common in this field of research. Specifically, we inverted what association rule mining was originally constructed to do, which we will discuss in this paper.

1.1 Metacognition and the Online Space

Especially in higher education contexts, where learning responsibilities often fall more on the student than the instructor, it is important to understand the behaviors related to students' academic Hannah Valdiviejas and Nigel Bosch "Using Association Rule Mining to Uncover Rarely Occurring Relationships in Two University Online STEM Courses: A Comparative Analysis" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 686 - 690

successes and failures. One behavior that oftentimes separates a successful student from a struggling student is metacognition [11]. Amongst metacognitive research, three main branches of metacognition have been distinguished: metacognitive knowledge, metacognitive monitoring, and metacognitive regulation [7]. For the sake of this research, we focus on metacognitive monitoring, as it is the critical point in order for metacognitive regulation to take place [3].

Metacognitive monitoring, or being conscious of what you do and do not know, is especially critical in online courses, because the burden of guiding and monitoring learning rests more on the student than in traditional learning environments [18]. To be a successful student in an online setting, where being self-regulated is crucial to academic success, the ability to be aware and strategize one's thinking is of the utmost importance. Students who accurately assess their mastery of a concept know how to take effective measures for studying that reflect this judgment of learning. This is called *calibration* and it can be detected through metacognitive monitoring [6]. Traditionally, metacognition in educational contexts has been analyzed according to interventions and surveys; however, this has been shown to be unreliable [15]. More often than not, metacognitive monitoring, a form of self-regulated learning, occurs subconsciously, making it difficult for students to accurately report this [9]. It is for this reason that we use an automatic metacognitive language detection tool [5], in order to avoid invalidities in traditional metacognition measurement.

1.2 Underrepresented Students in STEM

In the United States, an important issue remains unsolved year after year: that is, the vast underrepresentation of African American, Hispanic, Native American, first-generation, and non-male students in STEM majors and careers [2]. As if this were not troublesome enough, with each vertical stage in the academic process, the underrepresentation of these students gets worse [2]. A large underrepresentation of these students, and in turn, a large overrepresentation of people who do not identify with these demographic markers poses a serious bias in the trajectory of the nation, with only a small and homogenous group of people controlling sectors of business and research that are the engines of the nation's economy and innovation [14].

With the concern for students underrepresented (UR) in STEM existing throughout these students' educational trajectory, we argue that much can be learned by examining behaviors related not just to what might impede, but also what might support, these students' success in their STEM college courses to later improve representation in STEM fields. As online education continues to grow [1], its flexibility has made it a very attractive option for underrepresented students in STEM [4]. While online education does offer many options and benefits that traditional face-to-face education does not,

it must not only improve access to college courses among traditionally underserved students, but it must also support the academic success of these students. The purpose of this investigation is to document and understand some of the affordances of the online context for UR-STEM students in online STEM college courses.

1.3 Association Rule Mining

Unlike correlation analysis, which is bivariate, association rule mining can discover relationships among multiple variables at the same time [17]. Specifically, association rule mining aims to find “if-then” rules of the variables, in the form of “antecedent \rightarrow consequence,” where antecedent and consequence are conditions that some variable(s) has certain value(s). While association rule mining is extremely useful for exploratory analyses of large data, researchers have only recently attempted to grapple with a main issue of this tool: its inability to catch important, yet uncommon association rules [15]. This shortcoming of association rule mining poses an obstacle.

A handful of prior association rule mining research endeavors have expressed concern and proposed methods to remedy this issue. For example, [13] proposed confabulation-inspired association rule mining for finding rare itemsets. [12] stressed the importance of high-utility infrequent itemsets in fields like biology, banking, retail, and market basket analysis because of how infrequent itemsets find the hidden rules of association among the data items. In their research, they propose a Utility Pattern Rare Itemset (UPRI) algorithm to handle these scenarios. In terms of educational data mining, [16] explains that researchers will likely only find normal behavior in association rule mining because that is the most frequent behavior. To remedy this issue, [16] developed a new algorithm based on the Apriori approach to mine fuzzy specific rare itemsets from quantitative data, consisting of sets of items that rarely occur in the database together.

The current study aims to bring awareness to using association rule mining to catch rules amongst an already known subset of the participants, within the large dataset, rather than first mining in order to *discover* a subset group of the data that has characteristics in common. In this particular case it is minoritized underrepresented STEM students within a normal STEM online course. We applied association rule mining to explore the associations among variables pertaining to these students. For example, a possible rule in this study might be “non-male \rightarrow no prior online experience.” That is, if the student is a non-male, they are likely to have no prior online experience. Given that association rule mining tends to find frequent itemsets, we propose a modified approach in order to answer our research questions.

We ask the following research questions (RQs):

RQ1. What fine-grained relationships amongst underrepresented STEM students, their demographic information, and their metacognitive language can be uncovered through association rule mining?

RQ2. Although created to find commonly occurring sets of rules, can association rule mining be used to find sets of rules in an uncommon population (underrepresented students in STEM), within a larger set of data?

2 METHOD

In order to answer our research questions, we used demographic information from students in two online STEM courses and discussion forum posts from the same two courses to uncover fine-grained relationships between online learning behavior and student demographic variables.

2.1 Participants and Data Source

2.1.1 Discussion Forum

We analyzed all forum posts (7,040) from 205 students from one (8-week) term of Course A as well as all forum posts (6,086) from 77 students from one (16-week) term of Course B at a large Midwestern public university in the United States. All prompts that corresponded to the forum posts were open-ended with much flexibility for students to answer. Data included all of the students’ discussion forum posts as well as their final course grades, which were provided to us by university data curators. Specifically, there were four levels of grades: A, B, C, and D or lower (we combined D and F grades to avoid identifying students from this small group). In both courses, forum participation was required as part of students’ participation grades. Students were required to regularly post questions they had, or to answer other students’ questions. Online forum activity was 25% of their grade for students in Course A and 5% of their grade for students in Course B.

We used the [5] metacognition tool in order to count metacognitive phrases spontaneously produced by the students in their forum posts. We used this count to relate evidence of self-regulated learning behaviors to students’ background information. This tool also categorizes metacognitive language as being positive or negative; however, for the sake of this study, we only used total count.

2.1.2 Participants

Table 1 describes students’ demographic characteristics. Note that the total number of students across the subsamples is greater than the total of all students because some students belonged to more than one group. We do not report intersectional group level findings of students who fit multiple UR categories, to protect students’ identities and comply with FERPA regulations.

2.2 Data Analysis

Association rule mining has been used in educational contexts to find out relationships between variables, particularly in datasets with many variables [10], like in the current datasets (e.g., ethnicity, prior online experience, ACT score (a standardized test used for college admissions in the United States), grades, metacognitive language count).

Initially we used association rule mining tool as it was intended to be used but only found obvious associations, like those who are STEM majors are likely to have prior subject experience, with none of them dealing with underrepresented students in STEM. This is because their actions were not frequent compared to those in the majority (i.e., STEM majors) and therefore did not get detected as association rules. The current study’s process of association rule mining was inverted, meaning that the minimum support and lift values were set low because the target population was vastly underrepresented in the dataset. This included taking the inverse of many dummy variables where the majority was reflected rather than the minority; for example, we changed the variable “STEM major” to “Non-STEM major” so that we were mining for rules associated with the minority rather than the majority and the unlikely versus the obvious likely. In other words, all of the variables were changed to reflect the minority rather than majority in order to avoid excluding uncommon associations in these courses, especially dealing with minority groups. Therefore, we were actually looking for sets of *less* likely associations, relative to the total amount of associations, rather than likely associations. To identify interesting rules, the FP-Growth algorithm was used with a minimum Support value of 0.10, because the minimum population size

of some underrepresented student category groups that we looked at (non-males, racial/ethnic minoritized students, and first-generation) were just above 15% of the total population. In other words, if the minimum Support value was set higher than 0.15, it would not capture any of the association rules of the target population and if the minimum Support value were set right at 0.15, it would only capture those association rules in which all of the students pertaining to a specific category exhibited a particular rule. We selected a maximum Lift value of .89 since we wanted to find rules that were not associated with each other. High association, or associations that occur more than expected, are indicated by a Lift value > 1 in traditional uses of association rule mining. Therefore, a Lift value < 1 translates to events that happened less than expected. Through trial and error, we discovered that a Lift value set any lower than 0.89 would be too general and would generate too many rules. A Lift Value set higher than 0.89 gets too close to a high association value, excluding too many rules related to the underrepresented population we were interested in. Rules satisfying the criteria are defined as “interesting” in the sense that they were less likely to happen.

3 RESULTS

3.1 Descriptive Statistics

205 students in Course A produced a total 11,417 metacognitive phrases in 7,007 forum posts. The average number of metacognitive words per student was 55.69 (SD = 24.18). The final exam score was out of 170 points, and scores were approximately normally distributed. The minimum score a student received on the final exam was 69.26 and the maximum was 180 (with extra credit). The 77 students in Course B produced a total of 475 metacognitive phrases and 1,939 forum posts. The mean number of metacognitive phrases per student was 6.17 (SD = 5.07). Table 1 shows a percentage breakdown of the variables used in association rule mining in order to conceptualize Support values. *URM* signifies underrepresented racial/ethnic minoritized students in STEM (African American, Hispanic, and/or Native American), *First-gen.* signifies first generation college student (neither parent completed a higher education degree), *No Prior OL* refers to a student having no prior experience with an online course, a higher poster is a student who posts more than the class average (34 for Course A and 13 for Course B), *Low Exam* refers to the student getting a score lower than the class mean, *Course Rep.* refers to students taking the course for a second time (repeating), and *Non-tr. Age* refers to students older than 22.

Table 1. Student breakdown of variables used

Course A	205 Students	Course B	77 Students
Non-males	25%	Non-males	47%
URM	15%	URM	19%
First-Gen	16%	First-Gen	22%
No Prior OL	25%	No Prior OL	29%
High Poster	45%	Course Rep.	19%
Low Exam	47%	Non-Tr. Age	31%

3.2 RQ Answers

Table 2 shows the association rules that were likely to take place, or associations with a Lift value > 1 and Table 3 shows the association rules with a Lift Value < 1 that were less likely than average to occur. The meaning of each variable follows that of Table 1. The new variables include *Low Total MC* which signifies the student

produced less metacognitive language than the average of that class, *High Total MC* phrases refers to students producing more than the average for that class, and prior subject experience refers to students who have had experience with their current course’s subject. The strongest associations have Lift values > 1.00 and the weakest association all have Lift values < 1.00 .

3.3 Likely Association Rules

The two rules from Course A in Table 2 involve the *likely* associations among variables. In particular, the rule “High poster \rightarrow Non-male and isolates a strong association regarding who, of the underrepresented students in STEM, is engaging most in beneficial educational behaviors like posting often. “First generation \rightarrow Low total metacognition” suggests that first-generation students are not engaging metacognition as much as their peers.

The last two rules from Course B in Table 2 involve *likely* associations. The rule “Non-male, Non-traditional age group \rightarrow Low grade” suggests that non-males who are older than 21 are likely to receive lower grades than their peers. The rule “URM \rightarrow More than 4 metacognitive comments, Low grade” indicates if a student identifies as a URM, they are likely to engage in high amounts of metacognitive language but receive a low grade.

Table 2. Likely associations (Lift > 1)

	Antecedent	Consequence	Support	Lift
Course A	High poster	Non-male	0.13	1.16
	First-Gen	Low total MC	0.10	1.15
Course B	Non-male, Non-tr. age	Low grade	0.12	1.50
	URM	High total MC, Low grade	0.07	1.66

3.4 Less Than Average Association Rules

The first four rules from Course A in Table 3 involve the *unlikely* associations among variables. These are not simply the inverse of the most likely rules, because the minimum Support value was not changed, only the Lift. The rules “High poster \rightarrow Low metacognition” and “High poster \rightarrow Low exam” suggest that students who post often rarely exhibit low amounts of metacognition and rarely get low exam grades. The next two rules, “Low total metacognition, Low Exam \rightarrow Prior subject experience” and “Low metacognition \rightarrow Non-male”, indicate that the relationships between low metacognitive language and low exam score are rarely found amongst students with prior subject experience and non-male.

Table 3. Unlikely Associations (Lift $< .89$)

	Antecedent	Consequence	Support	Lift
Course A	High Poster	Low MC	0.09	0.55
	High Poster	Low Exam	0.09	0.70
	Low MC, Low Exam	Prior Subject Experience	0.06	0.73
	Low MC	Non-male	0.09	0.86
Course B	First-Gen, URM	No prior OL	0.06	0.79
	High total MC	Course repeat, Low grade	0.15	0.84
	First-Gen, Non-male	High total MC	0.19	0.87

The next three rules in Table 3 are unlikely associations from Course B. The rule “First generation, URM → no prior online experience” describes that if a student identifies as first-generation and as an URM, they are likely to have prior online experience. The rule “More than 4 metacognitive phrases → Course repeat, Low grade” indicate that it is unlikely for students to have negative educational outcomes if they are engaging in high amounts of metacognitive language. Lastly, the rule “First generation, Non-male → More than 4 metacognitive comments”, suggesting if a student is a first-generation and a non-male, it is highly unlikely that they are engaging in a high amount of metacognitive language production.

4 DISCUSSION

Based on the association rule mining analysis that was performed on data from an online Course A, there is evidence that suggests increased posting in this online course is associated with beneficial educational outcomes, like engaging in metacognitive learning strategies and obtaining a high exam grade. A more obvious rule uncovered through this analysis is that prior subject experience is also associated with beneficial educational outcomes. Some insight that rule mining provided about this course is that non-male students, although underrepresented in STEM, generally did well in this course while first-generation students did not fare as well.

Association rule mining also uncovered important information about students in Course B. A stark difference from Course A is that non-male students did not do as well in this course as in Course A. In Course B, being a non-male older than 22 years old was associated with getting a lower grade in the course. Being a non-male in general as well as being a first-generation college student was associated with uttering the least number of metacognitive phrases of all groups compared (gender, first-generation, and URM).

Underrepresented racial/ethnic minoritized students were the most likely group of students, amongst those compared, to produce metacognitive language; however, being a minoritized student was still associated with getting a lower grade in the course. This is an interesting finding because in Course A, the production of metacognitive language was positively related to course outcome however, in Course B it was not. Through association rule mining it is seen that the more metacognitive phrases a student produced, they less likely they were to display non-beneficial educational behaviors (i.e., repeat the course or receive a low grade).

Perhaps the most interesting finding of this analysis is that underrepresented racial/ethnic minoritized and first-generation college students were very likely to have prior online experience, but only for one course. Initially, before mining for association rules, we thought that a possible factor exacerbating the STEM achievement disparity was the digital divide, or the lack of experience that certain populations have with technology [8]. However, there is evidence that this is not the case. Along with research explaining that online education is an attractive option for underrepresented students [19], we see it is likely that underrepresented students have had prior experience with online education. Knowing this, educational researchers could hone in on this advantageous likelihood of experience with online courses to help lessen the underrepresentation of these students in STEM. The fact that this finding was only present in one course and not the other entertains explanations related to how there might be underlying similarities amongst students related to the types of courses they take, even within the STEM discipline.

4.1 Implications

Right now is a crucial time in higher education because of the apparent transition into more of an online state than ever before. We also know that online education is an attractive option for underrepresented students in STEM for various reasons (e.g., flexible class time). That being said, much work needs to be done in understanding academic outcomes in online education, especially for student underrepresented in STEM, because although it has great positive potential it also has the potential to worsen the lack of certain students in STEM majors and field.

The current study also indicates that association rule mining can, in fact, be used in other ways that it was not intended for, and in this case, to find commonly occurring sets of rules in an uncommon population (URMs), within a larger set of data. This opens the possibility for association rule mining to become a prevalent tool to be used among education researchers, especially to generate hypotheses about intersectional relationships that traditional statistical analyses might not uncover.

4.2 Future Directions

Association rule mining is intended to find variables that have strong associations to each other, in order to single out patterns not obvious by simply looking at the data. Using association rule mining was an issue when analyzing uncommon or non-majority populations, and therefore uncommon categories in the dataset, because the data miner has to take the inverse of what association rule mining was constructed to do. It is for this reason that we promote new algorithms or new ways of dealing with specific-rare itemsets, keeping in mind nuanced approaches that might be easier to use for educational researchers who are not entirely familiar with data mining techniques. Also, algorithms for rule mining that are specifically tailored to analyze unlikelihood or even likelihood but in minority subsets of data within the larger dataset would be very useful for reliable results and interpretation as well as facility in usage of educational data mining techniques. Future studies could also include extending these methods to more courses with varied demographics to determine the generalizability of using association rule mining in this way.

5 CONCLUSION

We took a novel approach to uncover relationships between student variables and course success by mining these variables for association rules in order to get a better understanding of the how UR-STEM students interact with online STEM courses.

We mined for unlikely as well as likely associations. We found interesting relationships that could prompt further analysis. These findings could be beneficial to an instructor, to provide clear direction about which students need direct help or additional resources, and thereby enhance positive outcomes in a course. These findings could also prove to be beneficial to online curriculum creators as well as university policy-makers because of specific information regarding an at-risk population (first-generation and racial/ethnic minoritized students) in the leaky STEM pipeline.

6 ACKNOWLEDGMENTS

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education through Grant R305A180211 to the Board of Trustees of the University of Illinois. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

7 REFERENCES

- [1] Allen, I.E. and Seaman, J. 2013. Changing course: Ten years of tracking online education in the United States. Sloan Consortium.
- [2] Estrada, M., Hernandez, P. R., & Schultz, P. W. (2018). A Longitudinal Study of How Quality Mentorship and Research Experience Integrate Underrepresented Minorities into STEM Careers. *CBE - Life Sciences Education*, 17(1).
- [3] Gourgey, A., (1998). Metacognition in basic skills instruction. *Instructional Science*, (1/2), 81
- [4] Gregory, C. B., & Lampley, J. H. (2016). Community college student success in online versus equivalent face-to-face courses. *Journal of Learning in Higher Education*, 12(2), 63-72.
- [5] Huang, E., Valdiviejas, H., & Bosch, N. (2019). I'm sure! Automatic detection of metacognition in online course discussion forums. In *Proceedings of the 8th International Conference on Affective Computing and Intelligent Interaction (ACII 2019)* (pp. 241–247). Piscataway, NJ:
- [6] Lingel, K., Lenhart, J., & Schneider, W. (2019). Metacognition in mathematics: do different metacognitive monitoring measures make a difference? *ZDM - Mathematics Education*, 51(4), 587–600.
- [7] Miller, T. M., & Geraci, L. (2011). Training Metacognition in the Classroom: The Influence of Incentives and Feedback on Exam Predictions. *Metacognition and Learning*, 6(3), 303–314.
- [8] Moore, R., Vitale, D., Stawinoga, N., & ACT Center for Equity in Learning. (2018). The Digital Divide and Educational Equity: A Look at Students with Very Limited Access to Electronic Devices at Home. *Insights in Education and Work*. In ACT, Inc. ACT, Inc.
- [9] Perry, N. E., & Winne, P. H. (2006). Learning from learning kits: Study traces of students' self-regulated engagements with computerized content. *Educational Psychology Review*, 18, 211-228.
- [10] Rojanavasu, P. (2019). Educational Data Analytics using Association Rule Mining and Classification. 2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON), Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON), 2019 Joint International Conference On, 142–145.
- [11] Rovers, S., Clarebout, G., Savelberg, H., de Bruin, A., van Merriënboer, J. (2019). Granularity matters: comparing different ways of measuring self-regulated learning. *Metacognition & Learning*, 14(1), 1–19.
- [12] Shrivastava, S., & Johari, P. K. (2016). Analysis on high utility infrequent ItemSets mining over transactional database. 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE International Conference On, 897–902.
- [13] Soltani, A., & Akbarzadeh-T., M. (2015). A new tree-based approach for evaluating rule antecedent constraint in confabulation based association rule mining. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 19(1), 1-14.
- [14] U.S. Bureau of Labor Statistics. (2019). Bureau of Labor Statistics. Consumer Price Index, 1–2.
- [15] Veenman, M. V. J., Van Hout-Wolters, B. H. A. M., & Af-flerbach, P. (2006). Metacognition and learning: Conceptual and methodological considerations. *Metacognition and Learning*, 1, 3-14.
- [16] Weng, C.-H. (2011). Mining fuzzy specific rare itemsets for education data. *Knowledge-Based Systems*, 24(5), 697–708
- [17] Xiao Hu, Weng-Lam Cheong, C., & Kai-Wah Chu, S. (2018). Developing a Multidimensional Framework for Analyzing Student Comments in Wikis. *Journal of Educational Technology & Society*, 21(4), 26–38.
- [18] Xu, D., & Jaggars, S. (2011). The effectiveness of distance education across Virginia's community colleges: Evidence from introductory college-level math and English courses. *Educational Evaluation and Policy Analysis*, 33, 360-377.
- [19] Xu, D., Jaggars, S. (2014). Performance gaps between online and face-to-face courses: Differences across types of students and academic subject areas. *Journal of Higher Education*, 85, 633-659.

Claim Detection and Relationship with Writing Quality

Qian Wan
Georgia State University
qwan1@gsu.edu

Scott Crossley
Georgia State University
scrossley@gsu.edu

Laura Allen
University of
New Hampshire
Laura.Allen@unh.edu

Danielle McNamara
Arizona State University
dsmcnama@asu.edu

ABSTRACT

In this paper, we extracted content-based and structure-based features of text to predict human annotations for claims and non-claims in argumentative essays. We compared Logistic Regression, Bernoulli Naive Bayes, Gaussian Naive Bayes, Linear Support Vector Classification, Random Forest, and Neural Networks to train classification models. Random Forest and Neural Network classifiers yielded the most balanced identifications of claims and non-claims based on the evaluation of accuracy, precision, and recall. The Random Forest model was then used to calculate the number, percentage, and positionality of claims and non-claims in a validation corpus that included human ratings of writing quality. Correlational and regression analyses indicated that the number of claims and the average position of non-claims in text were significant indicators of essay quality in the expected direction.

Keywords

argument mining, claim detection, essay quality, natural language processing, automated essay evaluation

1. INTRODUCTION

Argumentative essays include many different discourse units including a thesis statement, main ideas (claims), supporting ideas, and a conclusion (Burstein et al., 2003). Since argumentative essays are important elements in the teaching and assessment of writing, various techniques have been used to identify discourse units including those based on natural language processing (NLP). NLP has been used to automatically identify discourse elements based on the linguistic features that comprise discourse. Previous studies have found that content (i.e., lexical, syntactic, and discourse indicators) and structural features (i.e., the positionality of tokens, sentences, and paragraphs) are effective in the identification of discourse elements (Burstein et al., 1998, 2001a, 2001b, 2003; Lawrence and Reed, 2015; Nguyen and Litman, 2015, 2016; Persing and Ng, 2015; Stab and Gurevych, 2014, 2017). However, most studies have extracted content features at the word-level (unigram) or bigram level (e.g., Stab and Gurevych, 2017), or used indicators that generally occur only as transitional markers either at the beginning or the end of sentences (e.g., Burstein et al., 1998). Less is known about how multi-word n-grams (bigrams and trigrams) and their associated part-of-speech (POS) tags can influence the accuracy of discourse unit identification. Meanwhile, few if any studies, have examined how normalized positions of

sentences in paragraphs and in text can predict claims. Lastly, while some studies (e.g., Klebanov et al., 2016) have examined relations between essay quality and the use of discourse structures, these studies have examined relatively small corpora (e.g., test sets of 40 essays) and have not focused on claims, an important discourse element.

2. PURPOSE STATEMENT AND RESEARCH QUESTION

In this study, we develop NLP approaches to automatically identify claims in structurally-annotated essays using n-grams and POS tags along with positionality data. We compared the identification accuracy of the derived NLP features using different machine learning models and examined the relations between the number (and percentage) of claims and non-claims, their positionality, and human ratings of argumentative essay quality. Two structure-annotated corpora from Stab and Gurevych (2014, 2017) were used as our training ($N = 329$) and testing ($N = 90$) sets. The model with the best performance was used to identify claims and non-claims in a corpus comprising 2269 argumentative essays that had been rated on writing quality. Finally, we conducted correlation and regression analyses to explore the relations between the variables. The research questions that guide this study are as follow:

1. To what extent do (1) the frequency of n-grams (bigrams and trigrams), (2) the frequency of part-of-speech (POS) n-grams (bigrams and trigrams), and (3) positional (structural) information of sentences predict whether the sentence is a claim or not?
2. What are the relations between the number, percentage, and positionality of predicted claims/non-claims in an essay and the quality of the essay?

3. METHOD

3.1 Data

Three corpora were used in the current study. A training and testing corpora were used to train and test the claim detection algorithm, respectively. The claim detection algorithm was then applied to a validation corpus of student essays to calculate the number, percentage, and positionality of claims and non-claims in each essay. The relations of these features to claims (and non-claims) and essay quality was then examined.

3.1.1 Training set

The training corpus was developed by Stab and Gurevych (2017) and was annotated with argument components ("major claim," "claim," and "premises") and the relationships between "premises" and "major claim" or "claims" ("attack" or "support"). The corpus contains 402 argumentative essays written by students on 341 different prompts (e.g. "Will computers replace human power in jobs" and "Should students be taught to compete or cooperate").

Qian Wan, Scott Crossley, Laura Allen and Danielle McNamara "Claim Detection and Relationship with Writing Quality" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 691 - 695

The essays were collected from an online writing forum where native and non-native speakers of English could post their argumentative essays and give feedback to each other to help improve writing quality. After removing 73 essays that were duplicated in the testing set, there were 329 essays in the training corpus.

Major claims referred to sentences that directly expressed the general stance of the author that was supported by additional arguments. Claims were the central component of an argument, and premises were reasons that were provided by the author for supporting or attacking a certain claim. Three non-native annotators participated in the annotation process. According to the original study, the overall inter-rater agreement among the three annotators was .72.

3.1.2 Testing set

The testing corpus contained 90 argumentative essays compiled by Stab and Gurevych (2014). The essays were originally collected from the same source as the training set and were annotated by three annotators using the same annotation guidelines as the training set. It is unknown if the same annotators were used. The reported inter-rater reliability was .68.

3.1.3 Validation set

We selected 2269 argumentative essays written by native speakers of English as our validation corpus. The essays were collected in the development of the Writing-Pal (McNamara et al., 2012) from individual participants who composed essays in response to 13 specific prompts. Most of the participants were students ranging in grade levels from 7th to 10th or first-year college students. The participants were asked to respond to a specific prompt, state the degree to which they agreed or disagreed with the statement, and provide supporting evidence and arguments to persuade the readers. The essays in the validation corpus were evaluated by human raters following the scoring rubric used in the SAT (a standardized test used for college admittance in the United States). The SAT rubric evaluated writing in terms of ideas, organization, style, and voice. Raters were asked to assign each essay a quality score between 0-6. Interrater-reliability was greater than Cohen's Kappa .60 and r .70. Averages were taken between the two raters. If two raters disagreed by greater than one point on the 6-point scale, they were asked to adjudicate the essay. The average score for the essays was 3.38 and the standard deviation was .91.

3.2 Algorithm Development

Data preprocessing, feature development, application of machine learning models, and the selection of those models were the four major steps in the development of the classification algorithms for the claims and non-claims. We report the first two major steps in the following section and report the application and selection of machine learning models in the results section.

3.2.1 Merge and build standardized structure-annotated sub-corpora

The training and testing corpus were annotated using a framework of three argumentative units ("major claim," "claim," and "premise"). However, in this study we are only interested in distinguishing claims from non-claims. Based on our focus, we merged the tags of "major claim" and "claim" and treated both of them as a larger category of claim. We treated any sentences in an essay that did not fall into the category of claim as a non-claim. We

then unified the formats of the two structural annotated corpora by tokenizing the essays into sentence and adding structural tags (claim or non-claim) for each sentence based on the annotation of the original corpora. Further, we extracted all claim sentences from the training corpus to build the claim sub-corpus and extracted all the non-claim sentences to build the non-claim sub-corpus.

3.2.2 N-gram and n-gram POS tokenization

In this study, all of the n-gram and POS n-gram features for model development were extracted only from the training corpus. After the claim and non-claim sub-corpora were built, a Python script was written to tokenize the sentences within each corpus into bigrams and trigrams. Thus, all of the n-grams were extracted on sentence instead of clause levels. Prior to n-gram tokenization, all punctuations within the sentences were removed. Then, all of the characters were set to lowercase and all extra blanks in the sentences were removed from the texts. Stop words (e.g., *of*, *a*, *and*, *the*) were not deleted from the text. The texts were not lemmatized or stemmed.

We used the NLTK (Natural Language Toolkit; Bird et al., 2009) to tokenize the claim and non-claim sub-corpora into bigram and trigram. After the n-gram tokenization, we used the NLTK part-of-speech tagger to label the word class of each word within each sentence in the claim and non-claim sub-corpora. The NLTK pos-tagger labels part-of-speech for each word based on Penn Treebank tagset (Marcus et al., 1993). Prior studies have shown that the overall accuracy of NLTK pos-tagger was 91.33% for Brown Corpus, 89.56% for Treebank Corpus, and 86.45% for NPS Chat Corpus (Yumusak et al., 2014). Once the POS-tagging was completed, we used the NLTK tokenizer to segment the POS-tagged corpora into part-of-speech bigrams and trigrams. For example, the following phrases *should be*, *would be*, *can be*, and *will be* would be converted to the same POS n-gram combination: MD (modal) + VB (verb base).

3.2.3 Normalized frequency and Keyness values

We calculated raw frequency and normalized frequency for each bigram, trigram, as well as POS bigram and trigram term in the training corpus (both claim and non-claim sub-corpora). In addition to raw and normalized frequency, keyness value of each n-gram and POS n-gram was also calculated based on the raw frequency data. Keyness value, based on log-likelihood values, provided evidence of whether n-grams and POS n-grams were more common in one corpus compared with the other corpus (Kilgariff, 2001).

The thresholds for log-likelihood was 3.84 (equivalent to $p < .05$). Specifically, for any n-gram or POS n-gram that appeared in both corpora, if the n-gram or POS n-gram had a log-likelihood value greater than 3.84, we considered it more likely to occur in one corpus over the other. In this study, we wrote a Python script to automatically calculate the Keyness values (log-likelihood values) for all n-grams or POS n-grams that could be found in both claim and non-claim sub-corpora based on Rayson and Garside (2000). In Table 1, we list the top n-grams and POS n-grams with highest keyness values found in claims and non-claims.

In total, we calculated the following indices in the training, testing, and validation corpus, respectively: (1) the frequency of significant n-grams (bigrams and trigrams) in the claims extracted from the training corpus in each sentence; (2) the frequency of significant n-grams in the non-claims extracted from the training corpus in each sentence; (3) the frequency of significant POS n-grams in the claims in each sentence; and (4) the frequency of significant POS

n-grams in the non-claims in each sentence. In this way, for each sentence in each corpus, we derived eight indices.

Table 1 Top n-grams with highest keyness values in claims and non-claims

Significant Bigrams in Claims	Keyness	Significant Bigrams in Non-claims	Keyness	Significant Trigrams in Claims	Keyness	Significant Trigrams in Non-claims	Keyness
in conclusion	223.06	for instance	51.01	i believe that	67.08	more and more	8.38
i believe	77.18	able to	16.84	in my opinion	56.66	some people think	6.52
to sum	60.44	to go	13.11	to sum up	56.02	are able to	5.92
i think	56.57	i had	10.96	my point of	38.96	to go to	5.48
sum up	56.15	who have	10.96	point of view	35.94	in order to	5.03
in my	51.99	if you	10.85	as far as	28.50	in the past	4.41
believe that	50.85	did not	10.27	i prefer to	26.42		
my opinion	48.89	go to	10.04	first of all	23.95		
i strongly	44.68	means that	8.98	agree with the	19.66		
agree that	37.86	it was	8.85	from my point	19.30		
Significant POS Bigrams in Claims	Keyness	Significant POS Bigrams in Non-claims	Keyness	Significant POS Trigrams in Claims	Keyness	Significant POS Trigrams in Non-claims	Keyness
NN VBP	43.72	NN VBD	98.47	NN VBP IN	77.74	VBD TO VB	29.46
VBP IN	33.53	PRP VBD	69.91	RB VBP IN	32.32	VBD DT NN	28.74
NN MD	24.30	VBD RB	51.71	JJ VBP VBN	26.42	NN VBD RB	22.04
RBR JJ	19.58	VBD TO	40.06	VBP IN DT	19.86	IN PRP VBD	18.97
NNS MD	17.46	VBD DT	34.17	NN RB VBP	19.78	NN VBD DT	15.51
VBZ RBR	16.51	VBD VBN	25.34	TO VB RP	17.45	NN VBD VBN	15.35
JJ VBP	15.03	RB VBD	22.53	NNS MD VB	15.50	DT NNS RB	15.35
IN VBG	13.77	PRP VBP	20.96	NN MD VB	15.05	DT NN NN	13.35
NNS VBZ	12.76	VBZ VBN	19.40	VBZ RBR JJ	13.68	NN NN VBD	13.16
MD VB	11.74	VBD JJ	18.28	JJ NN VBZ	13.44	VBD IN NN	12.97

3.2.4 Positional data for sentences

Beyond n-gram patterns, studies have shown that in argumentative or academic writing, sentence position is an indicator of the structural function of the sentence (e.g., Burstein et al., 1998, 2001a; Biber et al., 2004). In this study, the following raw and normalized positional variables for each sentence in an essay were calculated as potential positional features: (1) the position of the sentence in the whole essay (e.g., if a sentence is the 5th sentence in the essay, the value of this variable would be 5); (2) normalized sentence position in the essay (i.e., equal to the value in [1] divided by the total number of sentences in the essay); (3) the position of the paragraph in which the sentence was located (e.g., if the sentence occurred in the 2nd paragraph of the essay, this value would be 2); (4) normalized paragraph position in the essay (i.e., equal to the value in [2] divided by the total number of paragraphs in the essay); (5) the position of the sentence in the paragraph where the sentence occurred (e.g., if the sentence was the 4th sentence in its paragraph, the value would be 4); and (6) the normalized position of a sentence in a paragraph (i.e., equal to the value in [5] divided by the total number of sentences in the paragraph).

3.3 Validation Study

Our second objective was to examine the relationship of the number/percentage of claims and positional data with the quality (human score) of the essay. To do so, the algorithm (from the final model) to predict the discourse type (claim or non-claim) was applied to each sentence of each essay in the validation corpus. We then calculated the percentages and average position of claim and non-claim sentences in each essay of the validation corpus and used these features to model essay quality to examine the following: (1) correlations between essay quality (represented by human holistic scores of the essays) and the number/percentage and positionality of claims/non-claims in the essay; and (2) the extent to which the number and percentage of claims/non-claims in an essay and sentence positionality predict its quality. In the regression analysis, the number of claims, the number of non-claims, the percentage of claims, the percentage of non-claims in an essay, and sentence positionality were included as the independent variables, while the human score of the essay served as the dependent variable. Prior to analyses, the human scores were checked for normality; multicollinearity ($r < .70$) across all independent variables was checked to ensure the variables developed were unique.

4. RESULTS

In the following sections, we report the results for feature selection, machine learning model selection, and the statistical analyses.

4.1 Feature Selection

As we have reported in the method section, we applied both content-based features and structure (position) based features to train the model.

Altogether, we had 17 features calculated at the sentence level. Six were structure (position) based features as reported in the method section: (1) the position of the sentence in the whole essay; (2) normalized sentence position in the essay; (3) the position of the paragraph in which the sentence was located; (4) normalized paragraph position in the essay; (5) the position of the sentence in the paragraph where the sentence occurred; and (6) the normalized position in paragraph. Eight of the features were content-based n-gram/POS n-gram frequency calculated based on sentence level. These features included: (1) the frequency of significant bigrams in claims; (2) the frequency of significant bigrams in non-claims; (3) the frequency of significant POS bigrams in claims; (4) the frequency of significant POS bigrams in non-claims; (5) the frequency of significant trigrams in claims; (6) the frequency of significant trigrams in non-claims; (7) the frequency of significant POS trigrams in claims; and (8) the frequency of significant POS trigrams in non-claims. The other three features were word counts, bigram counts, and trigram counts of the sentence.

Before moving forward to build the model, we conducted correlational analyses to remove highly correlated variables. The results of this analysis indicated that the position of the sentence in the essay was highly correlated with normalized sentence position in the essay ($r = .85, p < .001$), with the position of paragraph in essay ($r = .91, p < .001$), and with normalized paragraph position in essay ($r = .83, p < .001$). Normalized sentence position in essay was also highly correlated with the position of the paragraph in essay ($r = .89, p < .001$) and normalized paragraph position in essay ($r = .94, p < .001$). Meanwhile, the position of paragraph in essay was highly correlated with normalized paragraph position in essay ($r = .92, p < .001$). Based on these results, we decided to remove the position of sentence in essay, the paragraph position in essay, and the normalized paragraph position from the independent variables.

For the structure-based features, only the frequency of significant POS trigrams had a strong correlation with the frequency of significant POS bigram ($r = .54, p < .001$). For the word and n-gram counts variables, since the variable word counts were highly correlated with bigram counts ($r = 1, p < .001$) and trigram counts ($r = 1, p < .001$), we decided to remove both of the latter variables and only keep the variable of word counts. After this process, 10 features remained for model development (see Table 2).

Table 2 Summary of structural and content-based features for model development

Category	Feature
Structure (positional) features	Normalized sentence position in the essay
	Normalized sentence position in the paragraph
Content-based features	Word counts of the sentence
	The frequency of significant bigrams in claims in the sentence
	The frequency of significant bigrams in non-claims in the sentence
	The frequency of significant POS bigrams in claims in the sentence
	The frequency of significant POS bigrams in non-claims in the sentence
	The frequency of significant trigrams in claims in the sentence
	The frequency of significant trigrams in non-claims in the sentence
	The frequency of significant POS trigrams in claims in the sentence

4.2 Model Selection

We built six different supervised machine learning models on our training data using six different classifiers. We then we used the six models to predict discourse types of sentences in our testing corpus. We evaluated the performance of the models using accuracy, precision, recall, and F1-score. Table 3 reports the performance of the classifiers on claim and non-claim identification in the test set. The Random Forest model was selected as the best model to predict the discourse type in the validation corpus.

Table 3 Performance of the multiple classifiers on claim detection in the test set

		TP	TN	FP	FN	Precision	Recall	F1-score	Accuracy
LR	Claim	347	629	445	161	0.44	0.68	0.53	0.62
	Non-claim	629	347	161	445	0.80	0.59	0.67	
	Macro Avg					0.62	0.63	0.60	
	Weighted Avg					0.68	0.62	0.63	
BNB	Claim	129	965	109	379	0.54	0.25	0.35	0.69
	Non-claim	965	129	379	109	0.72	0.90	0.80	
	Macro Avg					0.63	0.58	0.57	
	Weighted Avg					0.66	0.69	0.65	
GNB	Claim	214	885	189	294	0.53	0.42	0.47	0.69
	Non-claim	885	214	294	189	0.75	0.82	0.79	
	Macro Avg					0.64	0.62	0.63	
	Weighted Avg					0.68	0.69	0.68	
LSVC	Claim	194	930	144	314	0.57	0.38	0.46	0.71
	Non-claim	930	194	314	144	0.75	0.87	0.80	
	Macro Avg					0.66	0.62	0.63	
	Weighted Avg					0.69	0.71	0.69	
RF	Claim	261	895	179	247	0.59	0.51	0.53	0.72
	Non-claim	895	261	247	179	0.78	0.83	0.80	
	Macro Avg					0.68	0.66	0.67	
	Weighted Avg					0.71	0.72	0.71	
NN	Claim	219	914	160	289	0.58	0.43	0.49	0.72
	Non-claim	914	219	289	160	0.76	0.85	0.80	
	Macro Avg					0.67	0.64	0.65	
	Weighted Avg					0.70	0.72	0.70	

Note: LR = Logistic Regression, BNB = Bernoulli Naive Bayes, GNB = Gaussian Naive Bayes, LSVC = Linear Support Vector Classification, RF = Random Forest, NN = Neural Network

4.3 Relationship between Essay Quality and Number of Claims

Spearman's correlations were computed among the number, percentage, and the average positionality of claims and non-claims and the human raters' holistic scores for each essay in the validation corpus. We included text length to assess if the raw scores highly correlated with the number of words in the essay (a strong predictor of essay quality). Correlational analysis indicated the number of predicted claims ($r = .35, p < .001$) and the average position of non-claims in text ($r = -.19, p < .001$) showed at least a small effect size ($r > .099$) with essay quality and were not strongly correlated with text length ($r < .70$). These variables were selected for inclusion in our regression analysis to predict essay quality scores. However, the percentage of predicted claims ($r = .08, p = .015$) and non-claims ($r = -.08, p = .015$) and the average position of claims ($r = .04, p < .001$) had weak correlations with essay quality.

A significant regression equation was reported ($R^2 = .132, F(2,2266) = 172.3, p < .001$). The model explained 13.2% of the variance of the human scores. Two significant predictors of essay quality were included in the model: number of claims ($\beta = .132, p < .001$) and the average position of non-claims in text ($\beta = -2.829, p < .001$).

5. CONCLUSION AND FUTURE WORK

In this study, we extracted content-based linguistic features and structure-based features to train and predict discourse types of claim and non-claim in argumentative essays. The average testing accuracy (F1) of the classifiers used in this study (Logistic Regression, Bernoulli Naive Bayes, Gaussian Naive Bayes, Linear

Support Vector Classification, Random Forest, and Neural Network) was around .69. This aligns with the accuracies reported in Stab and Gurevych (2017) to a degree. In their work, they reported F1 scores from an SVM classifier for major claims, claims, and premises using structural, lexical, contextual, syntactic, discourse markers, and embeddings features. Their F1 scores for these features in tandem were .77. F1 scores in isolation were .59 for lexical features, .60 for contextual features, .39 for syntactic features, .52 for discourse features, and .75 for structural features. These results seem to indicate that the individual content-based features (lexical, syntactic, indicator, and contextual features) might have encountered an upper limit in terms of the accuracy of identification if other features were not combined. The accuracy of the identification of claims in our study also seems to support this interpretation.

In terms of application, we found that the number of claims and the average position of non-claims in text were indicators of essay quality. A significant regression model was found to predict holistic human scores based on these variables. The model explained 13% of the variance in the human scores.

To improve the accuracy of classification, we are planning to implement a classifier with more diverse features from a contextual and discourse perspective including contextual, discourse, syntactic, and lexical features. We presume this will increase accuracy based on findings from Stab and Gurevych (2017) who showed that the combination of all features increased their accuracy. We also intend to investigate the relationship between argumentation elements from a broader view by including more argumentation elements such as major claims, primary claims, counterarguments, rebuttals, and conclusions. Further, we plan on annotating the relationships between these discourse elements and build models to automatically identify the discourse elements as well as their functional relationships.

In the current study, we have demonstrated the usefulness of content and structural features in automated claim detection and explored the relations between the number and positionality of claims and writing quality. Our findings can positively supplement existing automated essay scoring (AES) and automated writing evaluation (AWE) systems and may provide implications for the teaching of argumentative essays.

6. ACKNOWLEDGMENTS

This research was supported in part by the Institute for Education Sciences (IES R305A180261), Office of Naval Research N00014-17-1-2300, the Bill & Melinda Gates Foundation, The Chan Zuckerberg Initiative, and Schmidt Futures. Ideas expressed in this material are those of the authors and do not necessarily reflect the views of our funders.

7. REFERENCES

- [1] Biber, D., Conrad, S. and Cortes, V., 2004. If you look at...: Lexical bundles in university teaching and textbooks. *Applied linguistics*, 25(3), pp.371-405.
- [2] Bird, S., Klein, E. and Loper, E., 2009. Nltk book.
- [3] Burstein, J., Braden-Harder, L., Chodorow, M., Hua, S., Kaplan, B., Kukich, K., Lu, C., Nolan, J., Rock, D. and Wolff, S., 1998. Computer analysis of essay content for automated score prediction: A prototype automated scoring system for GMAT analytical writing assessment essays. *ETS Research Report Series*, 1998(1), pp.i-67.

- [4] Burstein, J., Kukich, K., Wolff, S., Lu, C. and Chodorow, M., 2001. Enriching Automated Essay Scoring Using Discourse Marking.
- [5] Burstein, J., Marcu, D. and Knight, K., 2003. Finding the WRITE stuff: Automatic identification of discourse structure in student essays. *IEEE Intelligent Systems*, 18(1), pp.32-39.
- [6] Burstein, J., Marcu, D., Andreyev, S. and Chodorow, M., 2001, July. Towards automatic classification of discourse elements in essays. In *Proceedings of the 39th annual Meeting on Association for Computational Linguistics* (pp. 98-105). Association for Computational Linguistics.
- [7] Kilgariff, A., 2001. Comparing corpora. *International journal of corpus linguistics*, 6(1), pp.97-133.
- [8] Klebanov, B.B., Stab, C., Burstein, J., Song, Y., Gyawali, B. and Gurevych, I., 2016, August. Argumentation: Content, structure, and relationship with essay quality. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)* (pp. 70-75).
- [9] Lawrence, J. and Reed, C., 2015, June. Combining argument mining techniques. In *Proceedings of the 2nd Workshop on Argumentation Mining* (pp. 127-136).
- [10] Marcus, M., Santorini, B. and Marcinkiewicz, M.A., 1993. Building a large annotated corpus of English: The Penn Treebank.
- [11] McNamara, D.S., Raine, R., Roscoe, R., Crossley, S.A., Jackson, G.T., Dai, J., Cai, Z., Renner, A., Brandon, R., Weston, J.L. and Dempsey, K., 2012. The Writing-Pal: Natural language algorithms to support intelligent tutoring on writing strategies. In *Applied natural language processing: Identification, investigation and resolution* (pp. 298-311). IGI Global.
- [12] Nguyen, H. and Litman, D., 2015, June. Extracting argument and domain words for identifying argument components in texts. In *Proceedings of the 2nd Workshop on Argumentation Mining* (pp. 22-28).
- [13] Nguyen, H. and Litman, D., 2016, August. Context-aware argumentative relation mining. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1127-1137).
- [14] Persing, I. and Ng, V., 2015, July. Modeling argument strength in student essays. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 543-552).
- [15] Rayson, P. and Garside, R., 2000, October. Comparing corpora using frequency profiling. In *Proceedings of the workshop on Comparing corpora-Volume 9* (pp. 1-6). Association for Computational Linguistics.
- [16] Stab, C. and Gurevych, I., 2014, October. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 46-56).
- [17] Stab, C. and Gurevych, I., 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3), pp.619-659.
- [18] Yumusak, S., Dogdu, E. and Kodaz, H., 2014. Tagging accuracy analysis on part-of-speech taggers. *Journal of Computer and Communications*, 2(4), pp.157-162.

VarFA: A Variational Factor Analysis Framework For Efficient Bayesian Learning Analytics

Zichao Wang¹, Yi Gu², Andrew S. Lan³, Richard G. Baraniuk^{1,4}

¹Rice University, ²Northwestern University, ³University of Massachusetts Amherst, ⁴OpenStax
jzwang@rice.edu, Yi.Gu@u.northwestern.edu, andrewlan@cs.umass.edu, richb@rice.edu

ABSTRACT

We propose VarFA, a variational inference factor analysis framework that extends existing factor analysis models for educational data mining to efficiently output uncertainty estimation in the model’s estimated factors. Such uncertainty information is useful, for example, for an adaptive testing scenario, where additional tests can be administered if the model is not quite certain about a students’ skill level estimation. Traditional Bayesian inference methods that produce such uncertainty information are computationally expensive and do not scale to large data sets. VarFA utilizes variational inference which makes it possible to efficiently perform Bayesian inference even on very large data sets. We use the sparse factor analysis model as a case study and demonstrate the efficacy of VarFA on both synthetic and real data sets. VarFA is also very general and can be applied to a wide array of factor analysis models. Code and instructions to reproduce results in this paper are available at <https://tinyurl.com/tvm4332>. An extended version of this paper is available at <https://arxiv.org/abs/2005.13107>.

1. INTRODUCTION

A core task for many practical educational systems is *student modeling*, i.e., estimating students’ mastery level on a set of skills or knowledge components (KC) [14, 6]. Such estimates allow in-depth understanding of students’ learning status and form the foundation for automatic, intelligent learning interventions. A fruitful line of research for student modeling follows the *factor analysis (FA)* approach. FA models usually assume that an unknown, potentially multi-dimensional student parameter, in which each dimension is associated with a certain skill, explains how a student answers questions and is to be estimated.

Most of the aforementioned FA models compute a single point estimate of skill levels for each student [13, 1, 3, 9, 5, 15]. Often, however, it is not enough to obtain mere point estimates of students’ skill levels; knowing the model’s uncertainty in its estimation is crucial because it potentially

helps improve the model’s performance and improve both students’ and instructors’ experience with educational systems. For example, in adaptive testing systems [4, 16], knowing the uncertainty in model’s estimation could help the model intelligently pick the next test items to most effectively reduce its uncertainty about estimated students’ skill levels. This will help to potentially reduce the number of items needed to have a confident, accurate estimation of the students’ skill mastery level, saving time for both students to take the test and instructors to have a good assessment of the student’s skills.

In this work, we propose VarFA, a novel framework based on *variational inference* (VI) to perform efficient, scalable Bayesian inference for FA models. The key idea is to approximate the true posterior distribution, whose costly computation slows down Bayesian inference, with a *variational distribution*. In addition, this variational distribution is very flexible and we have full control specifying it, allowing us to freely use the latest development in machine learning, e.g., deep neural networks (DNNs), to design the variational distribution that closely approximates the true posterior. Thus, we also regard our work as a first step in applying DNNs to FA models for student modeling, achieving efficient Bayesian inference (enabled by DNNs) without losing interpretability (brought by FA models). We demonstrate the efficacy of our framework on three real data sets, showcasing that VarFA substantially accelerates classic Bayesian inference for FA models with no compromise on performance.

2. BACKGROUND

We first set up the problem and review related work. Assume we have a data set $\mathbf{Y} \in \mathbb{R}^{N \times Q}$ organized in matrix format where N is the total number of students and Q is the number of questions. This is a binary students’ answer record matrix where each entry y_{ij} represents whether student i correctly answered question j . Usually, not all students answer all questions. Thus, \mathbf{Y} contains missing values. We use $\{i, j\} \in \Omega_{\text{obs}}$ to denote entries in \mathbf{Y} , i.e., the i -th student’s answer record to the j -th question, that are observed.

We are interested in models capable of inferring each i -th student’s skill mastery level that can accurately predict the student’s answers given the above data. These models are often evaluated on the prediction accuracy and whether the inferred student skill mastery levels are easily interpretable and educationally meaningful. We now review factor anal-

Zichao Wang, Yi Gu, Andrew Lan and Richard Baraniuk
"VarFA: A Variational Factor Analysis Framework For Efficient Bayesian Learning Analytics" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 696 - 699

ysis models (FA), one of the most widely adopted and successful methodologies for the student modeling task.

Many FA models, despite differences in their respective mathematical formulae, modeling assumptions and the available auxiliary data used, can be unified into a canonical formulation below

$$\mathbb{P}(y_{ij} = 1) = \sigma(\mathbf{c}_i^\top \mathbf{m}_j + \mu_j), \quad (1)$$

where $\mathbf{c}_i \in \mathbb{R}^K$, $\mathbf{m}_j \in \mathbb{R}^K$ and $\mu_j \in \mathbb{R}$ are factors whose dimension, interpretations and subscript indices depend on the specific instantiations of the FA model. We will use this general formulation in the rest of this paper. Usually, FA models obtains a point estimate of \mathbf{c}_i , \mathbf{m}_j and μ_j . We will show next how to obtain uncertainty estimation of these variables of interest.

3. VARFA: A VARIATIONAL INFERENCE FACTOR ANALYSIS FRAMEWORK

The core idea of VarFA follows the variational principle, i.e., we use a parametric variational distribution to approximate the true posterior distribution. VarFA is highly flexible and efficient, making it suitable for large scale Bayesian inference for FA models in the context of educational data mining. In this current work, we focus on obtaining credible interval for the student skill mastery factor \mathbf{c}_i 's as a first step of VarFA. Extension to VarFA to full Bayesian inference for all unknown factors is part of an ongoing research; see 5 for more discussions.

Now, we explain in detail how to apply variational inference for FA models for efficient Bayesian inference. Because the posterior distribution is intractable to compute, we approximate the true posterior distribution for \mathbf{c}_i 's with a parametric variational distribution

$$p(\mathbf{C}|\mathbf{Y}, \mathbf{M}, \boldsymbol{\mu}) \approx q_\phi(\mathbf{C}|\mathbf{Y}) = \prod_{i=1}^N q_\phi(\mathbf{c}_i|\mathbf{y}_i), \quad (2)$$

where ϕ is a collection of learnable parameters that parametrize the variational distribution and \mathbf{y}_i is all the answer records by student i . Notably, we have removed the dependency of the variational distribution on ψ and θ so that the variational distribution is solely controlled by the variational parameter ϕ . Thus, the design of the variational distribution is highly flexible. All we need to do is to specify a class of distributions and design a function parametrized by ϕ to output the parameters of q_ϕ . Common in prior literature is to use a Gaussian with diagonal covariance for q_ϕ :

$$q_\phi(\mathbf{c}_i|\mathbf{y}_i) = \mathcal{N}(\mathbf{u}_i, \text{diag}(\mathbf{v}_i)), \quad (3)$$

where its mean and variance $[\mathbf{u}_i^\top, \mathbf{v}_i^\top]^\top = f_\phi(\mathbf{y}_i)$. We can use arbitrarily complex functions such as a deep neural network for f_ϕ as long as they are differentiable. With the above approximation, Bayesian inference turns into an optimization problem under the variational principle, where we now optimize a lower bound, known as the evidence lower bound (ELBO) [2], of the marginal data log likelihood.

We form the following optimization objective to estimate ϕ

Table 1: Student answer prediction performance comparing VarFA to SPARFA-M on Assistment, Algebra and Bridge data sets. \uparrow and \downarrow denote higher and lower is better, respectively.

(a) Assistment		
Metric	Algorithm	
	SPARFA-M	VarFA
ACC \uparrow	0.7074 \pm 0.0044	0.7101 \pm 0.0048
AUC \uparrow	0.756 \pm 0.048	0.7635 \pm 0.0036
F1 \uparrow	0.7746 \pm 0.0029	0.7765 \pm 0.0014
Run time (s) \downarrow	5.3319 \pm 0.2774	6.9167 \pm 0.1074

(b) Algebra		
Metric	Algorithm	
	SPARFA-M	VarFA
ACC \uparrow	0.7735 \pm 0.0037	0.7774 \pm 0.0031
AUC \uparrow	0.8137 \pm 0.003	0.8245 \pm 0.002
F1 \uparrow	0.8465 \pm 0.0021	0.8486 \pm 0.001
Run time (s) \downarrow	8.464 \pm 0.4568	10.3335 \pm 0.4435

(c) Bridge		
Metric	Algorithm	
	SPARFA-M	VarFA
ACC \uparrow	0.8492 \pm 0.0016	0.8468 \pm 0.0016
AUC \uparrow	0.837 \pm 0.0024	0.8419 \pm 0.0028
F1 \uparrow	0.9121 \pm 0.0005	0.912 \pm 0.0009
Run time (s) \downarrow	15.6048 \pm 0.7314	15.8558 \pm 1.046

and θ :

$$\hat{\theta}, \hat{\phi} = \underset{\theta, \phi}{\operatorname{argmin}} -\mathcal{L}_{\text{ELBO}}(\phi, \theta) + \lambda \mathcal{R}(\theta), \quad (4)$$

where $\theta = \{\mathbf{m}_1, \dots, \mathbf{m}_Q, \mu_1, \dots, \mu_Q\}$ and $\mathcal{R}(\theta)$ is a regularization term. That is, we perform VI on the student factor \mathbf{c}_i 's and MLE inference on the remaining factors denoted as θ .

4. EXPERIMENTS

We demonstrate the efficacy of VarFA variational inference framework using the sparse factor analysis model (SPARFA-M) as the underlying FA model. On three real-world data sets, we demonstrate that 1) VarFA predicts students' answers more accurately than SPARFA-M; 2) VarFA can output the same insights as SPARFA-M, including point estimate of students' skill levels and questions' associations with skill tags; 3) VarFA can additionally output meaningful uncertainty quantification for student skill levels, which SPARFA-M is incapable of, without sacrifice to computational efficiency. Note that SPARFA-B can also compute uncertainty for small data sets but fails for large data sets due to scalability issues and thus we do not compare to SPARFA-B for real data sets. The code along with instructions to reproduce our experiments can be downloaded from <https://tinyurl.com/tvm4332>.

Data sets. We perform experiments on three large-scale, publicly available, real educational data sets including AS-

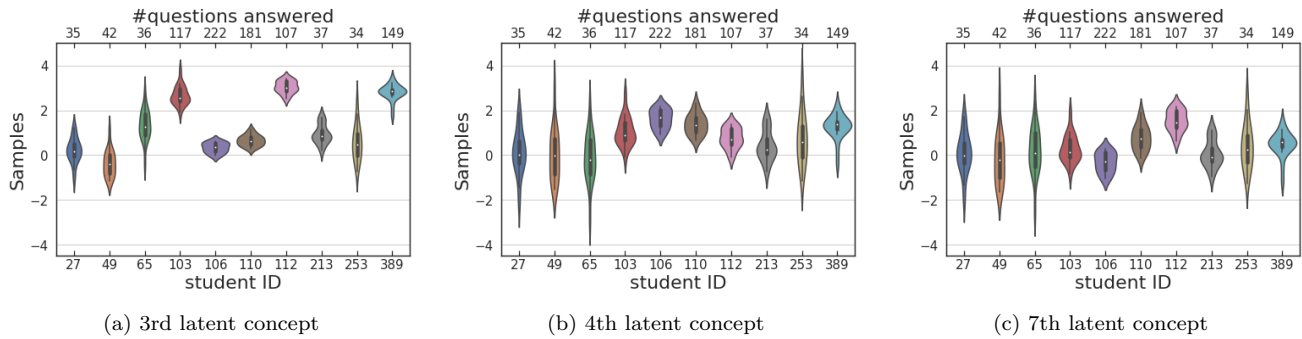


Figure 1: Violin plot showing the mean and standard deviation of the estimated skill mastery levels on 10 selected students on the 3rd, 4th and 7th latent skills that VarFA computes. In each sub-figure, bottom and top axes respectively shows student IDs and top axis shows the number of questions each student answered.

SISTments 2009-2010 (Assistment) [7], Algebra I 2006-2007 (algebra) [10] and Bridge to Algebra 2006-2007 (bridge) [11, 12]. The details of the data sets, including data format and data collection procedure can be found in the preceding references.

Results: Performance Comparison. Table 1 shows the average performance on the test set of each data set comparing VarFA and SPARFA-M for all three data sets and additionally run time. We can see that VarFA achieves slightly better student answer prediction on most data sets and on most metrics. Table 1 also shows the run time comparison between VarFA and SPARFA-M; see the last row in each sub-table. We see that both inference algorithms have very similar run time, showing that VarFA is applicable for very large data sets. Notably, VarFA achieves this efficiency while also performing Bayesian inference on the student knowledge level factor.

Results: Bayesian Inference With VarFA. We now illustrate VarFA’s capability of outputting credible intervals using the Assistment data set. Fig. 1 presents violin plots that show the sampled student latent skill levels for a random subset of 10 students. Plots 1a, 1b and 1c shows the inferred students ability for the 3rd, 4th and 7th latent skill dimension. In each plot, the bottom axis shows the student ID and the top axis shows the total number of questions answered by the corresponding student. For each student, the horizontal width of the violin represents the density of the samples; the skinnier the violin, the more widespread the samples are, implying the model’s less certainty on its estimations.

Results in Fig. 1 confirms our intuition that the more questions a student answers, the more certain the model is about its estimation. For example, students with ID 106, 110 and 389 answered 222, 181 and 149 questions, respectively, and the credible intervals of their ability estimation is quite small. In contrast, students with ID 27, 49 and 65 answered far less questions and the credible intervals of their ability estimation is quite large. This result implies that VarFA outputs sensible and interpretable credible intervals.

Results: Post-Processing for Improved Interpretability.

SPARFA assumes that each student factor \mathbf{c}_i identifies a multi-dimensional skill level on a number of “latent” skills (recall that we use 8 latent skills in our experiments). As mentioned earlier, these latent skills are not interpretable without the aid of additional information. To improve interpretability, [8] proposed that, when the skill tags for each question is available in the data set, we can associate each latent skill with skill tags via a simple matrix factorization. Then, we can compute each students’ mastery levels on the actual skill tags.

We again use the Assistment data set for illustration. We compute the association of skill tags in the data set with each of the latent skills and show 4 of the latent skills with their top 3 most strongly associated skill tags. We can see that each latent skill roughly identify the same group of skill tags. For example, latent skill 4 clusters skill tags on statistics and probability while latent skill 7 clusters skill tags on geometry. Thus, by simple post-processing, we obtain an interpretation of the latent skills by associating them with known skill tags in the data.

We can similarly obtain VarFA’s estimations of the students’ mastery levels on each skill tags through the above process. In Fig. 2, we compare the predicted mastery level for each skill tag (only for the questions this student answered) with the percent of correct answers for that skill tag. Blue curve shows the empirical student’s mastery level on a skill tag by computing the percentage of correctly answered questions belonging to a particular skill tag. Orange curve shows VarFA’s estimated student mastery level on a skill tag, normalized to range [0, 1]. Even though the two curves show different numeric values, they nevertheless demonstrate similar trends, showing that the predictions reasonably match our intuition about student’s skill mastery levels.

5. CONCLUSIONS AND FUTURE WORK

We have presented VarFA, a variational inference factor analysis framework to perform efficient Bayesian inference for learning analytics. VarFA is general and can be applied to a wide array of FA models. We have demonstrated the effectiveness of our VarFA using the sparse factor analysis (SPARFA) model as a case study. We have shown that VarFA can very efficiently output interpretable, education-

Table 2: Illustration of the estimated latent skills with the their top 3 most strongly associated skill tags in the Assistentment data set. The percentage in the parenthesis shows the association probability (summed to 1 for each latent skill). We see that the tagged skills associated with each estimated latent skill form intuitive and interpretable groups.

Latent Skill 1	Latent Skill 3
Division Fractions (29.1%) Least Common Multiple (18.1%) Write Linear Equation from Ordered Pairs (17.8%)	Conversion of Fraction Decimals Percents (7.3%) Addition and Subtraction Positive Decimals (6.8%) Probability of a Single Event (5.7%)
Latent Skill 4	Latent Skill 7
Pattern Finding (17.4%) Histogram as Table or Graph (11.3%) Percent Of (10.5%)	Volume Sphere (13.4%) Volume Cylinder (10.4%) Surface Area Rectangular Prism (10.2%)

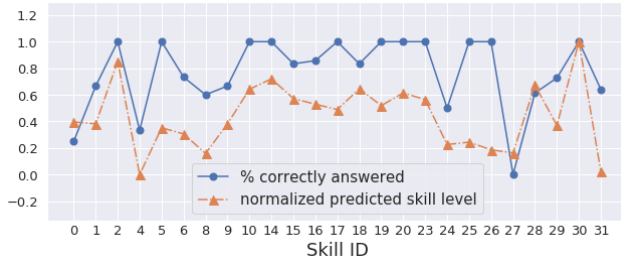


Figure 2: Comparison between the estimated skill mastery levels using VarFA's predictions and using empirical observations for student with ID 110.

ally meaningful information, in particular credible intervals, much faster than classic Bayesian inference methods. Thus, VarFA has potential application in many educational data mining scenarios where efficient credible interval computation is desired, i.e., in adaptive testing and adaptive learning systems. We have also provided open-source code to reproduce our results and facilitate further research efforts.

Acknowledgement

This work was supported by NSF grants CCF-1911094, IIS-1838177, IIS-1730574, DRL-1631556, IUSE-1842378; ONR grants N00014-18-12571 and N00014-17-1-2551; AFOSR grant FA9550-18-1-0478; and a Vannevar Bush Faculty Fellowship, ONR grant N00014-18-1-2047.

6. REFERENCES

- [1] T. A. Ackerman. Using multidimensional item response theory to understand what items and tests are measuring. *Applied Measurement in Education*, 7(4):255–278, 1994.
- [2] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.
- [3] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis – a general method for cognitive model evaluation and improvement. In M. Ikeda, K. D. Ashley, and T.-W. Chan, editors, *Proc. Intelligent Tutoring Systems*, pages 164–175, 2006.
- [4] H.-H. Chang, Z. Ying, et al. Nonlinear sequential designs for logistic item response theory models with applications to computerized adaptive tests. *The Annals of Statistics*, 37(3):1466–1488, 2009.
- [5] M. Chi, K. R. Koedinger, G. J. Gordon, P. Jordon, and K. VanLahn. Instructional factors analysis: A cognitive model for multiple instructional interventions. 2011.
- [6] K. Chrysafiadi and M. Virvou. Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications*, 40(11):4715–4729, 2013.
- [7] N. T. Heffernan and C. Heffernan. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24:470–497, 2014.
- [8] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk. Sparse factor analysis for learning and content analytics. *Journal of Machine Learning Research*, 15:1959–2008, 2014.
- [9] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis – a new alternative to knowledge tracing. In *Proc. Artificial Intelligence in Education*, page 531–538, NLD, 2009. IOS Press.
- [10] J. Stamper, A. Niculescu-Mizil, S. Ritter, G. Gordon, and K. Koedinger. Algebra i 2006-2007. *Development data set from KDD Cup 2010 Educational Data Mining Challenge*, 2010.
- [11] J. Stamper, A. Niculescu-Mizil, S. Ritter, G. Gordon, and K. Koedinger. Bridge to algebra 2006-2007. *Development data set from KDD Cup 2010 Educational Data Mining Challenge*, 2010.
- [12] J. C. Stamper and Z. A. Pardos. The 2010 kdd cup competition dataset: Engaging the machine learning community in predictive learning analytics. 2016.
- [13] W. J. van der Linden and R. K. Hambleton. *Handbook of modern item response theory*. Springer Science & Business Media, 2013.
- [14] K. VanLehn. Student modeling. *Foundations of Intelligent Tutoring Systems*, 55:78, 1988.
- [15] J.-J. Vie and H. Kashima. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proc. AAAI Conference on Artificial Intelligence*, volume 33, pages 750–757, 2019.
- [16] D. Yan, A. A. Von Davier, and C. Lewis. *Computerized multistage testing: Theory and applications*. CRC Press, 2016.

Next-Term Grade Prediction: A Machine Learning Approach

Audrey Tedja Widjaja, Lei Wang, Truong Trong Nghia,
Aldy Gunawan, Ee-Peng Lim
Singapore Management University
80 Stamford Road
Singapore, 178902

{audreyw, lei.wang.2019, ntruongtrong, aldygunawan, eplim}@smu.edu.sg

ABSTRACT

As students progress in their university programs, they have to face many course choices. It is important for them to receive guidance based on not only their interest, but also the “predicted” course performance so as to improve learning experience and optimise academic performance. In this paper, we propose the next-term grade prediction task as a useful course selection guidance. We propose a machine learning framework to predict course grades in a specific program term using the historical student-course data. In this framework, we develop the prediction model using Factorization Machine (FM) and Long Short Term Memory combined with FM (LSTM-FM) that make use of both student and course attributes as well as past student-course grade data. Our experiment results on a real-world data of an autonomous university in Singapore show that both methods yield better prediction accuracy than the baseline methods. Our methods are also robust to handle cold start courses with the average prediction error can be as low as three quarter grade difference from the ground truth.

Keywords

Grade prediction, factorization machine, long short term memory

1. INTRODUCTION

Predicting student grades has recently gained attention as it benefits not only students, but also instructors [3]. Students face many course courses in every new term. They need some guidance based on their “predicted” performance in future courses so as to improve their course selection and overall academic performance. Instructors, on the other hand, can also adjust their course delivery methods to the predicted student grade performance.

We consider a university setting where students are required to choose courses at the beginning of each program term.

Audrey Tedja Widjaja, Lei Wang, Nghia Trong Truong, Aldy Gunawan and Ee-Peng Lim "Next-Term Grade Prediction: A Machine Learning Approach" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 700 - 703

The predicted grades of the selected courses is then evaluated against the grades received at the end of that term. This task is called the *next-term student grade prediction* and it requires the past student-course grade data to provide useful features to predict grades of courses taken in the following term.

Our next-term student grade prediction task is different from the previous student grade prediction works [2, 3] which focused on predicting grades of a calendar term where students from different admission years are predicted together. Since different program terms are included in the prediction task, it is difficult to train the model to specialize on courses in the specific program term of the students.

In this paper, we develop FM and long short term memory combined with FM (LSTM-FM) models that are trained on student’s program terms instead of calendar terms. The proposed models are evaluated on a real-world data collected from an autonomous university in Singapore. We further make use of both static and dynamic student and course attributes to derive features that improve the prediction results. Additionally, our proposed models could perform well on predicting both existing and cold-start courses.

2. PROBLEM FORMULATION

Given a set of students $S = \{s_1, s_2, \dots, s_{|S|}\}$, where each student belongs to a certain cohort, denoted by $cohort(s_i)$ (i.e. batch of students admitting to the university in the same year). To graduate from their programs, students must complete $T = \{t_1, t_2, \dots, t_{|T|}\}$ program terms and register one or more courses in each program term. Let $C = \{c_1, c_2, \dots, c_{|C|}\}$ be the set of all courses taken by students from S . We denote the grade obtained by student s_i in course c_j by $g_{i,j} \in \{A+, A, \dots, F\}$. Our task is then to predict $g_{i,j}$ for every student s_i from a target student cohort S in a target program term t_k for every course students have registered in the program term t_k . We assume that the course grades for earlier program term(s) by the same students are available, and the course grades for students from previous cohorts in the earlier and target terms can be observed.

We define the feature representation of a student-course pair (s_i, c_j) as a feature vector $X_{i,j}$. A prediction model for the above problem is thus a function $F: X \rightarrow Y$ where $Y \in \mathbb{R}^2$. F is learned from a training data (t_k, X^{trg}, Y^{trg}) . For each

	Cohort 1	Cohort 2	Cohort 3	Cohort 4
Program term 1	Y^{trg}			Y^{test}

	Cohort 1	Cohort 2	Cohort 3	Cohort 4
Program term 1	X^{trg}			X^{test}
Program term 2	Y^{trg}			Y^{test}

	Cohort 1	Cohort 2	Cohort 3	Cohort 4
Program term 1	X^{trg}			X^{test}
Program term 2				
Program term 3	Y^{trg}			Y^{test}

Figure 1: Training and testing instances for program term-specific grade prediction involving data from cohorts 1 to 3 as training, and data from cohort 4 as test.

student $s_i \in S$, $Y_{i,j}^{trg}$ is unknown for courses c_j 's registered by s_i during the target program term t_k . For each student s_i of earlier cohorts, $Y_{i,j}^{trg} = g_{i,j}$ for courses c_j 's registered by s_i in the target program term t_k . For all students, $X_{i,j}^{trg}$ are features derived from student s_i and course c_j using data from earlier program terms. The testing data $(t_k, X^{test}, Y^{test})$ consists of $X_{i,j}^{test} = X_{i,j}^{trg}$ and $Y_{i,j}^{test} = g_{i,j}$ when s_i received the grade $g_{i,j}$ in the program term t_k .

Figure 1 illustrates the training and testing instances of the next term grade prediction for students of cohort 4 in target program terms 1 to 3. For target program term 3 (see the last table of the figure), the training data include the student-course data of students from cohorts 1 to 3. The feature representation of a student-course pair is derived from program terms 1 to 2 of these students, or from the non-program term student and course attribute data (e.g., student education background, course major, etc.).

This program term-specific grade prediction approach is more intuitive than previous works that focused on the grade prediction for students taking courses in the same calendar term which could involve different program terms for students from different cohorts [2, 3]. Since student grades of different program terms refer to different sets of courses, our problem definition and solution approach ensure that dyad features and ground truth labels for the testing data of a target program term follow the same data distribution as that of the training data.

3. DATASET AND FEATURES

3.1 Dataset Description

The dataset was collected from an autonomous university in Singapore that covers four consecutive cohorts (2011- 2014) of undergraduate students from the same degree program. Students are required to complete 8 program terms.

Table 1 shows the dataset statistics. It consists of 618 students and 691 courses. In total, we have 19,655 student-course pairs that involve grades, known as the student-course dyads. Students from cohort 4 are used as the test cohort to allow more data to be used in training. The university implements 12 grading letters that are mapped to numeric values for grade prediction as follows. A+, A, A-, B+, B, B-, C+, C, C-, D+, D, and F are mapped to 4.3, 4.0, 3.7, 3.3, 3.0, 2.7, 2.3, 2.0, 1.7, 1.3, 1.0, and 0.0 respectively.

Table 1: Dataset Statistics

	Cohorts				Total
	1	2	3	4	
Num. Students	115	145	157	201	618
Num. Courses	169	160	170	192	691
Num. Dyads	3748	4471	4850	6586	19,655

Table 2: Student-Course Dyads of Target Cohort 4 (CSS: cold start students, CSC: cold start courses, NCS: non-cold start dyads)

Program term	#dyads	#NCS	CS	
			#CSC	#CSS
t_1	986	0	0	986
t_2	955	952	3	0
t_3	856	850	6	0
t_4	919	907	12	0
t_5	801	789	12	0
t_6	704	677	27	0
t_7	699	676	23	0
t_8	666	638	28	0

Cold start dyads. The cold start student-course dyads of a target program term are ones with new students or courses with respect to the program term. They do not appear in the training set, but appears in the testing set. As shown in Table 2, program term t_1 sees all cold start dyads with new students (denoted by CSS). The other program terms however hardly encounter new students. Dyads involving cold start/new courses (denoted by CSC) are relatively fewer as not many new courses are introduced in each program term. Most of the new courses are observed in the program terms t_6 to t_8 , the last 3 terms of the program. The other dyads are the non-cold start (NCS) dyads.

3.2 Student-Course Features

We consider five categories of features for representing the student-course dyads (s_i, c_j) :

Static student features. These are features of a student which do not change with time as they are not associated with any target program term, such as student's *major*, *gender*, *alma_mater*, and *cohort*.

Dynamic student features. These are student features derived from the data and their values may vary in different target program terms. These features are particularly useful to determine the latest performance and academic load of the student, such as student's average grade in the previous program term (*lterm_gpa*) and up to previous program term (*lterm_cum_gpa*), number of credit units (CUs) a student received up to previous program term (*total_chrs*) and registered in the target program term (*term_chrs*), average CUs per program term taken by a student (*speed*), number of courses taken by a student in every course discipline up to target program term (*disc_distrib*), relative CUs gained by a student compared to all students in the same cohort (*rel_total_chrs*), and relative *lterm_cum_gpa* of a student compared with that of the cohort (*rel_lterm_cgpa*).

Static course features. These are features of a course c_j that do not change with time: course's discipline (*disc*), CUs (*chrs*), and level (*clevel*).

Dynamic course features. These are features of a course c_j that change with time: instructor of c_j (*iid*), number of students taking c_j in the target program term (*num_enrolled*) and in all previous program terms (*total_enrolled*), average grade (*term_cgrade*) and grade distribution (*term_dgrade*) obtained by students of the previous cohort when they took c_j in the target program term, average grade (*lterm_cum_cgrade*) and grade distribution (*lterm_cum_dgrade*) obtained by students of the same and previous cohorts when they took c_j in any program terms in the past.

Student-course interaction features. As we know which student s_i takes which course c_j in the target program but not the grade, we can exploit this information to derive some features that capture the *indirect* interaction between s_i and c_j for us to determine if s_i will perform well in c_j . We derive *rel_ctype* that measures the program term s_i registered for c_j relative to the program term other students of the same cohort taking c_j . We also derive *disc_grade* which is the average grade obtained by s_i when taking any courses sharing the same course discipline as c_j in the previous terms.

4. PROPOSED METHODS

Two methods are proposed for the next-term grade prediction task, namely, **Factorization Machine (FM)**, and **Integrated Long and Short Term Memory with FM (LSTM-FM)**. The former is often used for recommendation tasks. The latter is a sequence model combined with FM to predict grades of courses in each program term.

4.1 Factorization Machine (FM)

To use FM for next-term grade prediction, our training data for predicting grades in a target term t_k is represented by a $N_{trg}^{dyads} \times p$ matrix, X , where N_{trg}^{dyads} represents the number of training dyads, $p = |S| + |C| + |F|$, and F represent the set of features. Each row $X(i, j)$ for dyad (s_i, c_j) consists of a one-hot vector of student ids, a one-hot vector of course ids, and the features representing the dyad (s_i, c_j) .

Model. FM captures both 1-way and 2-way interactions between all features using factorized interaction parameters, as formulated below.

$$\hat{Y}_{i,j} = w_0 + \sum_{k=1}^p w_k X_{i,j,k} + \sum_{k=1}^p \sum_{k'=1}^p X_{i,j,k} X_{i,j,k'} \sum_{f=1}^k v_{k,f} v_{k',f}$$

where w_0 captures the global intercept and together with the $\sum_{k=1}^p w_k X_{i,j,k}$ serves as a basic linear regression model. The last part contains all pairwise interactions of the X features, which is modeled as a factorized parameterization $\sum_{f=1}^k v_{k,f} v_{k',f}$.

4.2 LSTM-FM Model

In LSTM-FM, we merge a sequence model with FM to both learn the sequence of grades received by a student and predict the grades in the target program term using the observed sequence as well as the feature interaction for the student-course dyads. The LSTM-FM framework (Figure 2) is decomposed into two main components: 1) *Input Layer* that utilizes bidirectional LSTM networks (Bi-LSTM) [1] to model the historical grades of a student and 2) *Interaction Layer* that employs interaction module similar to FM in order to model features interactions. The returned value is

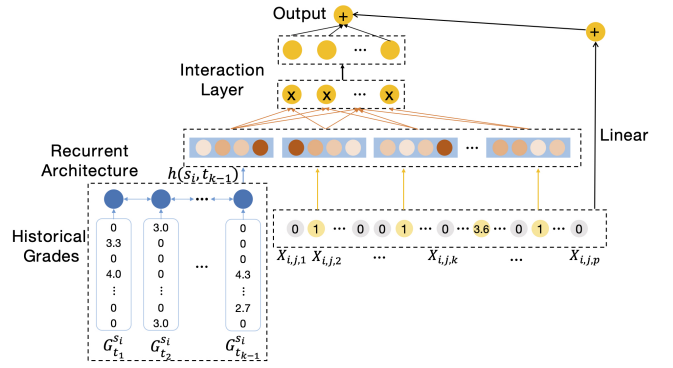


Figure 2: LSTM-FM framework

then transformed into the predicted grade by using 2-layer feed-forward networks with layer normalization [4].

As there can be a number of courses taken by the student in the same program term, we define $G_{t_k, c_j}^{s_i}$ as a $|C|$ dimensional vector keeping the grade score of student s_i gets for course c_j in program term t_k . We then use historical courses-grades of student s_i , $G_{t_k, c_j}^{s_i}$'s, for terms t_1, \dots, t_{k-1} to learn the hidden states using Bi-LSTM. We subsequently concatenate the hidden states $\vec{h}(s_i, t_{k-1})$ and $\vec{h}(s_i, t_{k-1})$ of the bi-LSTM into $h(s_i, t_{k-1})$ which is fed to the interaction layer with the (s_i, c_j) 's features to predict $G_{t_k, c_j}^{s_i}$.

5. EXPERIMENTS

5.1 Evaluation Metrics

Root mean squared error (RMSE) and mean absolute error (MAE) are used to evaluate the accuracy of different grade prediction methods as formulated below. The grades need to be converted to numerical values before using the two metrics. For both RMSE and MAE, the error is defined by the difference between the predicted grade and the actual grade. RMSE is appropriate to penalize methods that yield large errors. MAE, on the other hand, provides the average difference between the predicted and actual grades. For example, for a given actual grade of A- (with numeric score = 3.7), an MAE of 0.3 suggests that the predicted grade differs from the actual grade by an average of half grade, say B+ (with score = 3.4) or A (with score = 4.0).

$$RMSE = \sqrt{\frac{\sum_{Y_{i,j}^{trg} \text{ is defined}} (\hat{Y}_{i,j}^{test} - Y_{i,j}^{test})^2}{|\{(i, j) | Y_{i,j}^{trg} \text{ is defined}\}|}}$$

$$MAE = \frac{\sum_{Y_{i,j}^{trg} \text{ is defined}} |\hat{Y}_{i,j}^{test} - Y_{i,j}^{test}|}{|\{(i, j) | Y_{i,j}^{trg} \text{ is defined}\}|}$$

5.2 Methods for Evaluation

We focus on evaluating FM and LSTM-FM with the features defined in Section 3. There are several variants for both depending on what features are used: FM and LSTM-FM without any features other than student id and course id are also included (**FM and LSTM-FM without features**),

Table 3: Overall Results (CSC: Cold Start Courses)

Method	All dyads		Dyads w/o CSC		Only CSC dyads	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
UR	1.710	1.382	1.735	1.404	1.696	1.378
GM	0.755	0.551	0.757	0.552	0.638	0.506
MoM	0.676	0.488	0.678	0.488	0.583	0.448
Without student-course features						
LR	0.628	0.456	0.629	0.455	0.577	0.434
FM	0.607	0.428	0.608	0.428	0.552	0.415
LSTM-FM	0.651	0.464	0.652	0.464	0.618	0.490
With all student-course features						
LR	0.629	0.457	0.630	0.459	0.585	0.446
FM	0.625	0.448	0.622	0.445	0.587	0.457
LSTM-FM	0.628	0.449	0.629	0.449	0.574	0.441
With selected student-course features						
LR	0.621	0.452	0.621	0.455	0.583	0.452
FM	0.594	0.425	0.594	0.428	0.601	0.450
LSTM-FM	0.603	0.437	0.603	0.436	0.606	0.476

FM and LSTM-FM with all features and FM and LSTM-FM with only selected features (Section 5.3).

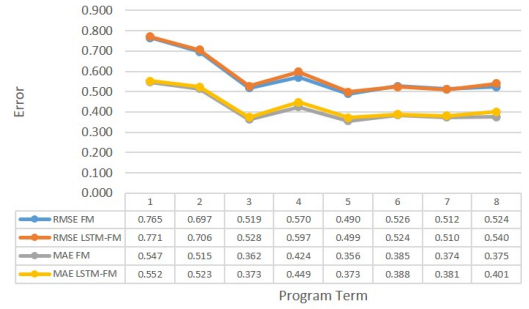
We include several baseline methods for comparison: **uniform random (UR)** that randomly predicts a grade score from interval $[0, 4.3]$, **global mean (GM)** that predicts a grade score using the average of all observed grades in the training set, **mean of means (MoM)** that returns the average of (a) the predicted grade score of GM; (b) the average observed grades of this student in the training set; and (c) the average observed grades of this course in the training set, and **linear regression (LR)** that uses the first two components of FM ($w_0 + \sum_{k=1}^p w_k X_{i,j,k}$) to predict a grade.

5.3 Prediction Results

The overall prediction results are summarized in Table 3. UR yields the highest error. With the use of historical data, GM can predict with smaller errors. MoM further reduces the prediction error with more information used. By implementing a traditional machine learning approach, LR, we can obtain lower prediction error. The results show that the historical data contribute to grade prediction accuracy, and it is worthwhile to explore more machine learning approaches to improve this grade prediction task.

We then analyse the results of our proposed methods. It is interesting to see that FM with only student id and course id predicts grades quite well. It is also applied to LSTM-FM although the latter has a larger error. FM (and LSTM-FM) with all features actually performs worse than the one without features. With selected features (by excluding *cohort*, *disc_distrib*, *iid*, *term_dgrade*, and *lterm_cum_dgrade*), both methods achieve the best results. The overall results show that the lowest error obtained by LR in every scenario is always higher than those of FM and LSTM-FM. This suggests that the 2-way interaction captured in both FM and LSTM-FM can improve prediction accuracy compared to LR that only captures linear model. The results so far are encouraging as an MAE of 0.425 is smaller than a $\frac{3}{4}$ grade difference. We evaluate the methods for dyads that do not involve CSC to see if they are able to improve prediction accuracy. Table 3 shows that CSC dyads do not make significant difference to the prediction results. This suggests that the methods are robust against CSC.

The prediction errors for each program term are illustrated in Figure 3. We observe that both FM and LSTM-FM have similar performance on predicting grades in every program


Figure 3: Prediction error per program term

term. The first two program terms t_1 and t_2 have relatively higher errors compared to the latter terms due to lesser training data. t_1 also handles grade prediction for cold start students. As the amount of training data increases, we notice a significant error improvement from term t_3 onwards. The error converges at term t_5 when the model has sufficient training data. For terms t_5 to t_8 , both methods can maintain the MAE to be below 0.401.

6. DISCUSSION AND FUTURE WORK

Based on the proposed framework in this paper, we plan to develop a grade prediction API for the university that can be used by both students and instructors. This may help students to select courses that are appropriate to enroll, given their performance in past terms. Instructors then may use this API to understand the class profile, see the predicted performance of their students and use this information to adjust class outline and delivery method. We plan to explore using course description and knowledge graph to improve prediction accuracy. More advanced deep learning models can also be introduced to explain the prediction results.

7. ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Singapore under its International Research Centres in Singapore Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

8. REFERENCES

- [1] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.
- [2] M. Sweeney, J. Lester, and H. Rangwala. Next-term student grade prediction. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 970–975, Oct 2015.
- [3] M. Sweeney, H. Rangwala, J. Lester, and A. Johri. Next-term student performance prediction: A recommender systems approach. *CoRR*, abs/1604.01840, 2016.
- [4] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu. On layer normalization in the transformer architecture. *CoRR*, abs/2002.04745, 2020.

Detecting Problem Statements in Peer Assessments

Yunkai Xiao [yxiao28],¹ Gabriel Zingle [gzingle],¹ Qinjin Jia [qjia3],¹ Harsh R. Shah [hshah3],¹

Yi Zhang [20171184],² Tianyi Li [tianyili],³ Mohsin Karovaliyya [mrkarova],¹

Weixiang Zhao [20172986],² Yang Song [songy],⁴ Jie Ji [20172986],⁵

Ashwin Balasubramanian [abalasu4],¹ Harshit Patel [hpatel24],¹

Priyanka Bhalasubramanian [pbhalas],¹ Vikram Patel [vpatel22],¹ and Edward Gehringer [efg]¹

¹ North Carolina State University, Raleigh, North Carolina 27695, USA [@ncsu.edu]

² Northeastern University, Shenyang, Liaoning 110819, China [@stu.neu.edu.cn]

³ Shanghai Jiao Tong University, Shanghai 201101, China [@sjtu.edu.cn]

⁴ University of North Carolina at Wilmington, Wilmington, North Carolina 28407, USA [@uncw.edu]

⁵ Southern University of Science and Technology, Shenzhen, Guangdong 518055, China [at@mail.sustech.edu.cn]

ABSTRACT

Effective peer assessment requires students to be attentive to the deficiencies in the work they rate. Thus, their reviews should identify problems. But what ways are there to check that they do? We attempt to automate the process of deciding whether a review comment detects a problem. We use over 18,000 review comments that were labeled by the reviewees as either detecting or not detecting a problem with the work. We deploy several traditional machine-learning models, as well as neural-network models using GloVe and BERT embeddings. We find that the best performer is the Hierarchical Attention Network classifier, followed by the Bidirectional Gated Recurrent Units (GRU) Attention and Capsule model with scores of %93.1 and %90.5 respectively. The best non-neural network model was the support vector machine with a score of 89.71%. This is followed by the Stochastic Gradient Descent model and the Logistic Regression model with 89.70% and 88.98%.

Keywords

Peer assessment, problem detection, text mining, text analytics, machine learning

1. INTRODUCTION

Peer assessment—students giving feedback on each other’s work—has been a common educational practice for at least 50 years [1, 2] It provides students more copious and rapid feedback than an instructor would give, as well as reactions from a more authentic audience (the student’s peers). By concentrating on a limited number of works, peers can produce assessments with similar validity and reliability to those of instructors, whose time is spread more thinly over many students’ submissions [3]. Students who perform peer

assessment show a substantial increase in performance [4]. Moreover, studies uniformly report that students learn more by being reviewers than they learn from the reviews they receive [5, 6, 7, 8].

The need for peer assessment was felt more acutely after the rise of massive open online courses (MOOCs). With students paying little to no fees, MOOCs are not able to hire enough staff to assess all submitted work. Thus, MOOCs rely heavily on peer assessment [9, 10].

For students to gain from peer assessment, students must take the process seriously. They must think carefully and metacognitively about the works they are reviewing. To foster an atmosphere where students assess conscientiously, the instructor must train the students in reviewing—and follow up by assessing how well they perform this task [11]. But instructor assessment of students’ reviewing suffers from the same shortcomings as instructor assessment of students’ submitted work: it consumes much instructor time, is likely to be rushed, and is mostly summative; that is it evaluates how well the students have done, but does not directly help them improve their reviewing. Thus, considerable research has looked at other methods for assessing review quality [12].

Fundamentally, the quality of a review is related to whether it identifies ways for the author to improve the work. Thus, the review should point out shortcomings or problems the reviewer perceives in the reviewed work. This paper describes several approaches to automatically identifying whether review *comments*, which are responses to individual rubric items, do point out (alleged) problems with the work.

2. RELATED WORK

Previous approaches to evaluating peer-assessment reviews include calibration [13, 10, 14], reputation systems [15, 16], “back-reviews” (rejoinders) [17], natural language processing [18, 19, 20], logistic regression [21], and neural-network techniques [22]. Peer assessment has much in common with peer review, as used to vet scientific work for publication. Hua et al. [23] used NLP to automatically detect arguments in these reviews. Negi [24] used several AI techniques to detect suggestions in product reviews. Space does not permit

Yunkai Xiao, Gabriel Zingle, Qinjin Jia, Harsh Shah, Yi Zhang, Tianyi Li, Mohsin Karovaliyya, Weixiang Zhao, Song Yang, Jie Ji, Ashwin Balasubramanian, Harshit Patel, Priyanka Bhalasubramanian, Vikram Patel and Edward Gehringer “Detecting Problem Statements in Peer Assessments” In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 704 - 709

elaboration of these methods, but a fuller discussion can be found in [our extended paper](#).

3. EXPERIMENTAL METHODOLOGIES

3.1 Data

The data used for our experiments comes from Expertiza [25], a peer-assessment platform for reviewing work developed by collaborative teams. For each review, the reviewer fills out a rubric, which consists of several criteria. Sample rubric items are, "How well does the code follow good Ruby and Rails coding practices?" "Is the user interface intuitive and easy to use?" Most criteria ask for a numeric rating as well as textual feedback. It is the textual feedback that we analyze in this work.

Our study is based on reviews of coding and documentation assignments from NCSU CSC 517, Object-Oriented Design and Development. To obtain labeled data for our research, we offered students a small amount of extra credit for tagging review comments they received, as either mentioning a problem or not. We spot-checked the student-assigned tags for the purpose of quality control. An example comment that does not mention a problem (tagged as 0) is, "The interface is easy to use and it is well described in the README file." One mentioning a problem (tagged as 1) is, "The implementation can only log one type of user on."

Several students had the opportunity to tag the same review comments. If multiple students tagged the same comment, inter-rater reliability (IRR) could be calculated. We used Krippendorff's α [26] as the metric for IRR. By dropping observations with conflicting tags, we have raised the Krippendorff's α associated with our dataset from 0.696 to 1.

The dataset was de-duplicated and balanced, resulting in a total of 18,354 observations. It was separated into training, validation, and testing sets in the ratio of 80:10:10. This split was used to find optimized hyperparameters with 5-fold cross-validation. Unless the dataset is large, the combination of observations used in the training and test sets can have an impact on how well the classifier performs. We compensated for this by using 20-fold cross-validation on our finalized classifiers with tuned hyperparameters and saving the resulting 20 scores for analysis.

3.2 Baseline Models

We set up our baseline using traditional machine-learning models, such as Support Vector Machine (SVM), SVM using Stochastic Gradient Descent (SGD), Multinomial Naïve Bayes (MNB), Logistic Regression (LR), Random Forest (RF), Gradient Boosting (GB), and AdaBoost (AB).

3.2.1 Input Embedding

The input to our baseline models was first processed by the TF-IDF vectorizer in scikit-learn [27]. TF-IDF vectorization is a common way to convert raw text and documents into embeddings suitable for machine-learning models. The vectorizer generates a document-vocabulary matrix for each of the documents (in our case, review comments that averaged 2.2 sentences per comment). Then, using inverse document frequency, it normalizes ("lowers") the weight of the words by checking how often a word appears in other documents

(comments, in this case). This helps lessen the impact of frequent yet unimportant words, so that common words like "the" that convey little semantic meaning do not affect the classification of a comment. The model architecture and dataflow for traditional classifiers is shown in Figure 1.

3.2.2 Support Vector Machine

Support vector machines are commonly used for classification in machine learning. A SVM establishes a decision boundary as well as a positive plane and a negative plane between classes. Statistical features for each review comment represented in TF-IDF-normalized vectors are put into the vector space for all comments, then the model learns a hyper-plane (support vector) to best divide them into two categories: comments containing problem statements, and comments without problem statements.

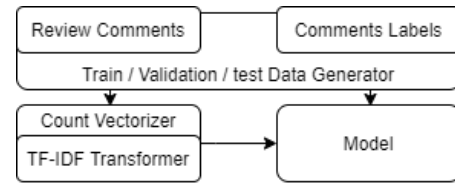


Figure 1: Data pipeline for machine learning model

3.2.3 SVM with Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) was developed early on and popularly adopted to optimize neural-network models [28], while applying SGD on linear classifiers is not unheard of. [29] We compared the performance of the SVM model with and without SGD. We applied a combination of L1 and L2 regularization to the loss function, with the hope of correcting over-fitting problems.

3.2.4 Multinomial Naïve Bayes

A naïve Bayes model assumes that each of the features it uses for classification is independent the others. To determine whether a review comment identifies a problem, the model examines the TF-IDF normalized word-count vectors for that comment, using the conditional probability of each of these features/vectors, and makes a judgment, based on conditional probabilities learned from the training set.

3.2.5 Logistic Regression

The logistic-regression (LR) classifier uses a regression equation to produce discrete binary outputs. Similar to linear regression, it learns the coefficients of each input feature through training; however it uses a logistic function instead of linear activation to determine the class to which an input belongs by fitting coefficients of each n-gram through comments in the given training set.

3.2.6 Random Forest

The Random Forest (RF) classifier is an ensemble method that fits multiple decision trees and uses averaging to improve the accuracy of predictions and to avoid over-fitting.

3.2.7 Gradient Boosting

Gradient boosting (GB) is an ensemble machine-learning algorithm that utilizes a number of weak models, such as small

decision trees. In training, these small decision trees are fitted in a negative gradient direction in order to reduce the loss calculated from the cost function.

3.2.8 AdaBoost

AdaBoost, or adaptive boosting, is a meta-algorithm that alters weights of entries for base models. When an entry is misclassified, the algorithm increases the weight of that entry and decreases the weights of entries that have been correctly classified. The algorithm terminates upon meeting the confidence threshold. Through doing this, the booster identifies the features that have greater impact on the results, and improves prediction accuracy.

3.3 Neural Network Models

Our other experiments use neural networks, and Keras [30] was the framework of choice for implementation. Compared with our baseline models, the input of each model is generated in two different ways: through a GloVe embedding and BERT embedding.

3.3.1 Input Embedding

Global Vectors for Word Representation, or GloVe embedding [31], is an embedding model that converts words into multidimensional vectors based on their meaning. Its function is similar to Word2Vec, which transforms words to embeddings in a limited vector space, though the underlying principle is different.

Bidirectional Encoder Representations from Transformers (BERT) is a multi-layer bidirectional transformer encoder [32] developed by Google. The BERT network we used in our experiment is published by Google and is pre-trained on Wikipedia and BooksCorpus data. We used the open-source project "Bert-as-service" to create sentence embeddings. Specifically, we limited the maximum sentence length to 25 words, and extracted embeddings with outputs from the second-last layer in the pretrained network. The Bert-Base-Uncased model [32] has 12 attention layers, and 768 neurons in each layer with 12 attention heads. Using this network has given us 768 dimensions as sentence embeddings. We also used a version with word level embeddings. Figure 2 demonstrates the model architectures in order of the next subsection.

3.3.2 Multilayer Perceptron

A multilayer perceptron (MLP) model [33] is a typical artificial neural network. It utilizes multiple layers of neurons, and uses back-propagation for training. Errors calculated by a loss function are propagated back through the layers using the chain rule of gradient descent derivation.

3.3.3 Convolutional Neural Network

A convolutional neural network (CNN) utilizes convolution kernels that pool data with a defined window size on given dimensions to generate summaries from input data [34].

When dealing with comment classification, this model uses convolutions on the feature dimension to reduce the complexity of each word vector, different dropout percentages, and pooling methods.

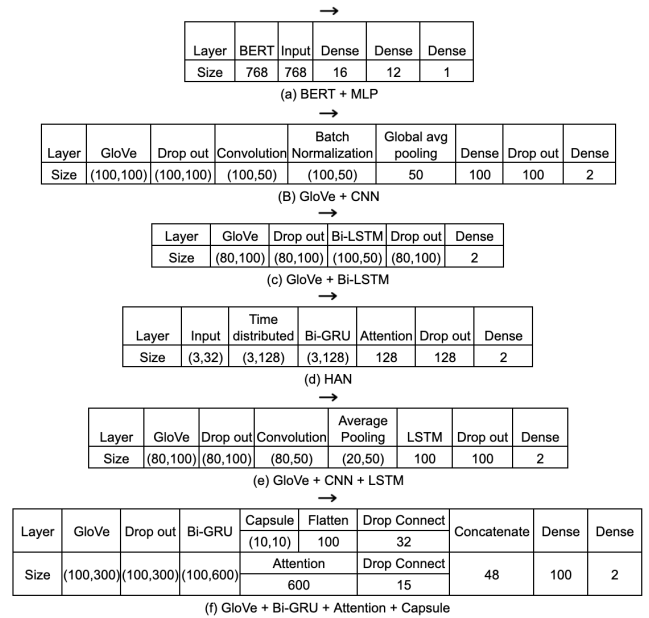


Figure 2: Data pipeline for neural network Models

3.3.4 Recurrent Neural Network

Recurrent neural networks (RNNs) are neural networks that take time-sequence information into consideration. For each time-step, the network takes the inputs and updates its internal memory cells with new information. Different RNN models implement memory updates differently. For example, long short-term memory (LSTM) networks not only remember inputs, but also "forget" unimportant information.

When we pass an embedded sentence to the network, each word is seen as an item emerging in one time step, and the sequence of words in a sentence becomes a sequence of vectors transitioning along with time steps. The neural network learns from the transition what information is important to keep and what is not, then applies the same judgment when a new sentence is given to it for classification.

Here we also implemented a GRU network and a bidirectional GRU network in parallel.

3.3.5 Hierarchical Attention Network

Hierarchical attention networks (HANs) are neural networks that take into consideration the document structure and sentence structure [35]. A document normally consists of a number of sentences, and a sentence is formed by a number of words. Not all sentences in a document are important to the classification of a document, and similarly, not all words are important for sentence-level classification. HANs utilize this information through attention layers that capture words and sentences that are important towards the classification.

In classifying comments, a HAN can capture information with greater impact on the results. For example in sample comment "The writeup does not include a Test Plan section," the words "does not include" contributes a lot more to implying there is a problem stated in this comment than other parts of the comment do.

3.3.6 CNN with Long Short Term Memory

Previous models showed that each type of the neural network or neural network layer could be efficient on specific tasks, for example CNN for dimension reduction and HAN for extracting words that are more important to the result. In this subsection we combine some models and explore the benefits of mixing different types of neural networks.

A model with CNN and LSTM layers is implemented in the hope of securing benefits from both models. With CNN as a dimension reducer, the LSTM layer might be able to find useful information from the aggregated features. Another attempt tests whether a CNN is needed to reduce dimensions, by removing it while boosting the performance of recurrent layer by putting it in a bidirectional wrapper.

4. EXPERIMENTAL RESULTS

Figure 3 displays a boxplot of the 20 f1-scores obtained using the traditional machine-learning classifiers and neural networks from the 20-fold cross validation. The lowest-performing classical machine learning classifiers, multinomial naïve Bayes and AdaBoost, achieved similar accuracy, with respective sample median f1-scores of 0.855 and 0.861. The gradient boosting and random-forest classifiers achieved sample median f1-scores of 0.870 and 0.871. The highest performing classifiers included logistic regression, stochastic gradient descent, and support vector machines. They achieved sample median f1-scores of 0.890, 0.897, and 0.897 respectively.

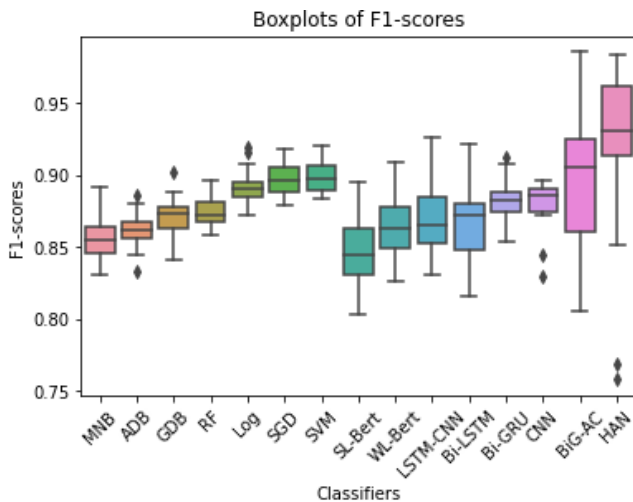


Figure 3: Models' F-1 Scores

These results show that classifiers can classify review comments as mentioning problems with an accuracy range of approximately 84% to 95%.

The HAN and BiGRU-Attn-Caps models that used GloVe embeddings achieved the best performance among all the models. The CNN model that used GloVe embeddings achieved the next best performance with a sample median f1-score of 0.886. The Bidirectional GRU had a very close sample median f1-score of 0.882, followed by the Bidirectional LSTM model with 0.872, then the LSTM CNN model at 0.865. The lowest-scoring models were the ones with word-level (WL-

Bert) and sentence-level (SL-Bert) BERT embeddings with sample median f1-scores of 0.862 and 0.844 respectively.

To gain insight into the phrases that contributed towards determining a suggestion, we extract coefficient weights of some features from two of the models. Table 1 displays a list of the logistic regression model's top 10 positive and negative features in determining if a comment has mentioned a problem in the author's work. The features that increase the likelihood that a comment will mention a problem (positive coefficients) include phrases that may constitute a suggestion by the reviewer. For instance, phrases such as "could", "should", "could have", and "more" indicate that the reviewer is likely giving advice to the author about improving the work, thus noting a problem by implication. Features with negative coefficients include phrases that likely demonstrate positive sentiment, such as "yes", "good", "well", and "great".

Table 1: Logistic Regression Coefficients

Coefficient	Value	Coefficient	Value
yes	-8.0233	not	10.5227
good	-3.9472	but	8.8498
and	-3.1690	however	7.8254
they have	-3.1193	more	6.2155
well	-3.0567	could	5.6703
yes the	-2.9953	should	5.3498
all the	-2.7422	would	5.0391
clearly	-2.6269	no	5.0183
project	-2.5331	missing	4.9864
passed	-2.4645	some	4.9160

Table 2 displays the stochastic gradient descent model's top 10 positive and negative features in determining if a comment mentioned a problem in the author's work. The coefficient values are lower than those of the logistic regression model, but they comprise similar positive and negative features.

Table 2: Stochastic Gradient Descent Coefficients

Coefficient	Value	Coefficient	Value
yes	-4.1029	however	6.5277
conflicts	-2.0396	not	6.4184
good	-2.0083	but	5.5175
apply	-1.7788	should	3.9721
complicated	-1.7785	could	3.9198
since	-1.6178	would	3.8352
sense	-1.6139	more	3.6346
required	-1.5925	missing	3.5942
passed	-1.5757	no	3.4112
project	-1.5637	except	2.9776

5. SUMMARY

We have marshalled a multitude of classifiers that can parse student peer-review comments for the detecting the mention of a problem. The HAN and BiGRU-Attn-Caps models performed the best among the neural network classifiers on this dataset, while the best traditional classifiers were the support vector machine and stochastic gradient descent models. The least effective classical models were the AdaBoost and multinomial naïve Bayes classifiers—the two that used the sentence and word level embeddings.

6. REFERENCES

- [1] Keith Topping. Peer assessment between students in colleges and universities. *Review of Educational Research*, 68(3):249–276, 1998.
- [2] Joanna Tai and Chie Adachi. 5 the transformative role of self-and peer-assessment in developing critical thinkers. *Innovative Assessment in Higher Education: A Handbook for Academic Practitioners*, 2019.
- [3] Keith J Topping. Peer assessment. *Theory into Practice*, 48(1):20–27, 2009.
- [4] Hongli Li, Yao Xiong, Charles Vincent Hunter, Xiuyan Guo, and Rurik Tywoniw. Does peer assessment promote student learning? a meta-analysis. *Assessment & Evaluation in Higher Education*, pages 1–19, 2019.
- [5] Kristi Lundstrom and Wendy Baker. To give is better than to receive: The benefits of peer review to the reviewer’s own writing. *Journal of Second Language Writing*, 18(1):30–43, 2009.
- [6] Yasemin Demiraslan Çevik. Assessor or assessee? investigating the differential effects of online peer assessment roles in the development of students’ problem-solving skills. *Computers in Human Behavior*, 52:250–258, 2015.
- [7] Lan Li, Xiongyi Liu, and Allen L Steckelberg. Assessor or assessee: How student learning improves by giving and receiving peer feedback. *British Journal of Educational Technology*, 41(3):525–536, 2010.
- [8] Esther Van Popta, Marijke Kral, Gino Camp, Rob L Martens, and P Robert-Jan Simons. Exploring the value of peer feedback in online learning for the provider. *Educational Research Review*, 20:24–34, 2017.
- [9] Judy Kay, Peter Reimann, Elliot Diebold, and Bob Kummerfeld. MOOCs: So many learners, so much potential ... *IEEE Intelligent Systems*, 28(3):70–77, 2013.
- [10] Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. Tuned models of peer assessment in MOOCs. *arXiv preprint arXiv:1307.2579*, 2013.
- [11] Xiongyi Liu and Lan Li. Assessment training effects on student assessment skills and task performance in a technology-facilitated peer assessment. *Assessment & Evaluation in Higher Education*, 39(3):275–292, 2014.
- [12] Edward F Gehringer. A survey of methods for improving review quality. In *International Conference on Web-Based Learning*, pages 92–97. Springer, 2014.
- [13] Orville L Chapman and Michael A Fiore. Calibrated peer reviewTM. *Journal of Interactive Instruction Development*, 12(3):11–15, 2000.
- [14] Yufeng Wang, Hui Fang, Qun Jin, and Jianhua Ma. SSPA: an effective semi-supervised peer assessment method for large scale MOOCs. *Interactive Learning Environments*, pages 1–19, 2019.
- [15] John Hamer, Kenneth TK Ma, and Hugh HF Kwong. A method of automatic grade calibration in peer assessment. In *Proceedings of the 7th Australasian Conference on Computing education-Volume 42*, pages 67–72. Australian Computer Society, Inc., 2005.
- [16] Kwangsu Cho and Christian D Schunn. Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. *Computers & Education*, 48(3):409–426, 2007.
- [17] Luca De Alfaro and Michael Shavlovsky. Crowdgrader: A tool for crowdsourcing the evaluation of homework assignments. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, pages 415–420. ACM, 2014.
- [18] Wenting Xiong and Diane Litman. Automatically predicting peer-review helpfulness. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 502–507. Association for Computational Linguistics, 2011.
- [19] Wenting Xiong, D Litmaan, and Christian Schunn. Natural language processing techniques for researching and improving peer feedback. *Journal of Writing Research*, 4(2):155–176, 2012.
- [20] Caroline Brun and Caroline Hagege. Suggestion mining: Detecting suggestions for improvement in users’ comments. *Research in Computing Science*, 70(79.7179):5379–62, 2013.
- [21] Huy Nguyen, Wenting Xiong, and Diane Litman. Instant feedback for increasing the presence of solutions in peer reviews. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 6–10, 2016.
- [22] Zhongcan Xiao, Chandrasekar Rajasekar, Ferry Pramudianto, Edward Gehringer, Vishal Chittoor, and Abhinav Medhekar. Application of neural-network models to labeling educational peer reviews. In *CSEDM 2018: Educational Data Mining in Computer Science Education Workshop*, Buffalo, NY, 2018. IEDMS.
- [23] Xinyu Hua, Mitko Nikolov, Nikhil Badugu, and Lu Wang. Argument mining for understanding peer reviews. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2131–2137, Minneapolis, MN, June 2019. Association for Computational Linguistics.
- [24] Sapna Negi and P Buitelaar. Suggestion mining from opinionated text. *Sentiment Analysis in Social Networks*, pages 129–139, 2017.
- [25] Edward F Gehringer. Expertiza: Managing feedback in collaborative learning. In *Monitoring and Assessment in Online Collaborative Environments: Emergent Computational Technologies for E-learning Support*, pages 75–96. IGI global, 2010.
- [26] Klaus Krippendorff. *Content analysis: An introduction to its methodology*. Sage Publications, 2013.
- [27] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.
- [28] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

- [29] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [30] François Chollet et al. Keras: Deep learning library for theano and tensorflow. *URL: <https://keras.io/k>*, 7(8):T1, 2015.
- [31] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [33] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.
- [34] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [35] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

An Empirical Analysis of Skewed Temporal Data for Distribution-based Course Similarity

Tao Xie
Faculty of Education
Southwest University
Chongqing, China
xietao@swu.edu.cn

Chaohua Gong
Faculty of Education
Southwest University
Chongqing, China
chaohua@swu.edu.cn

Geping Liu
Faculty of Education
Southwest University
Chongqing, China
gepingli@swu.edu.cn

ABSTRACT

Time series data that exhibit skewed distribution is a common and important issue related to advanced model adoption, which however may be mis-specified when the data become extremely large and completely stochastic. This study adopted an experience-to-model approach in order to address the data skewness problem in educational data mining and in parallel explain the practical pedagogical meaning of data distribution patterns. To do this, we first specified a proper analysis granularity with respect to temporal data and provided evidence of its non-normality, and finally handled the skewness by correlating it to gaussian mixture models. We performed a scalable model by adaptively selecting the parameters and discussed the similarity measure based on probability density distribution.

Keywords

Data skewness, temporal pattern, data transformation, e-learning

1. INTRODUCTION

In recent years, big data in education is becoming a new driving force and playing an increasingly important role in educational research and practice[1]. The mining of big data in education is beneficial for educators and organizations to understand the learning patterns of students, optimize curriculum design, gain insight into student characteristics, provide high-quality educational decisions, and finally improve students' academic standards[2-4].

One of main challenges, however, is that the data is not always of normal distribution, which makes many standard approaches limited and corresponding results not robust. Many studies either ignored the existence of this challenge or simplified the assumptions of research conditions. Pearson correlation, for example, is used for testing linear dependence between a couple of variables assuming the data is small and has a normal shape. But the model can be easily mis-specified because the feasibility of this assumption is weakened when the data become extremely large and completely stochastic. Besides, many scholars use machine learning algorithms to classify the time series data without paying much attentions to the data distributions, leading to seriously inaccurate results due to the fact that the performances of classifiers

are subject to the data presented to it during training session and many attributes of the data are not balanced[5]. The imbalanced data distribution is usually described by a skewness coefficient in statistics representing an asymmetry from the mean of a data distribution. Time series data that exhibit skewed behavior is a common and important issue related to advanced model adoption of educational data mining[6].

This study adopts an experience-to-model approach in order to address the data skewness problem in educational data mining and in parallel explain the practical meaning of data distribution patterns. To do this, we first summarized coarse-grained observations on temporal data that collected from online courses, and then discussed the non-normality of time series data in education based on carefully selected granularity. Finally, we provided a tutorial to handle the skewness by correlating it to gaussian mixture models.

2. RELATED WORK

Temporal data has been studied in many research domains, while there is only a little literature has been documented in the educational domain. The current research on educational temporal data has mainly focused on online courses and metacognition. For example, authors in[7] proposed a temporal modeling approach for students' dropout prediction in MOOCs, authors in[8] mined temporal characteristics of learning behaviors from e-learning systems, and authors in[9] obtained sequential and temporal characteristics of self and socially regulated learning. Many of these research omitted the distribution assumption of data samples, which may lead to improper explanations related to statistical values. As is known, the Gaussian distribution is well known and widely applied by assuming that the aggregate effect of many individual independent components tends to be distributed with symmetrical bell curve. However, the use of Gaussian-based statistics can result in substantial error if problems are involved a lot of skewed data[10]. The assumption of homogeneity of variance indicates that the variance of the variable remains constant over the observed range, which may not be the truth in most research scenarios. Although the current statistical software packages provide tools to test the normality assumptions, and a lot of literature have documented to use multiple regression model and ANOVA model for modest violations to these assumptions[11]. A more effective way, however, is to transform data to improve normality of independent variables when substantial non-normality is present. Data transformations can improve normality of a distribution and equalizing variance in quantitative analysis of data. For this reason, this study will conduct experiments based on this approach. In previous works, the transformation approaches include adding constants, square root transformation, log transformation, scales, inverse transformation, arcsine

Tao Xie, Chaohua Gong and Geping Liu "An Empirical Analysis of Skewed Temporal Data for Distribution-based Course Similarity" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 710 - 714

transformation and Box-Cox transformation[12]. Many of these approaches are showing good properties of distribution symmetry.

Besides, the data skewness problem has been extensively studied in communities of time series mining. It is claimed that the skewness has significantly influence on the performance of algorithmic tasks[13], where the authors have to detect the degree of skewness to determine the characteristics of the dataset distribution. To explore such influences, the relationship between data skewness and accuracy of data mining models has been examined in[14] and[15]. Because the irregular sampling of data sets is often encountered in time series, the measures of skewness are also of great interest and have become especially important while conducting data cleaning and data preprocessing. Therefore, authors in[16] devised an approach to transform the time series segments to produce new ones so that the new ones can be analyzed using standard methods, which is in essence consistent with data transformation techniques as stated above. This study uses quadratic square root approach on temporal data and compares the difference in the developed metrics between original and transformed data on online courses.

3. DATA

3.1 Data Collection

There are 14 million video-viewing logs collected from 57,717 students. We choose the top 7 video courses with the most in-course interactions, which are respectively Introduction to Mao Thought (MS), Political Economy (PE), Linear Algebra (LA), Enterprise Financial Management (EF), Marketing (MM), Microcomputer Principle and Interface (MI), and Health Assessment (HA). Finally, we keep information of 7,341 students. One-way ANOVA shows that the differences between groups in terms of the continuous variables are statistically significant at the 0.05 level ($df = 6, p = 0.00$) and the differences between groups in terms of the age and video-viewing time have statistical significance at the 0.05 level ($df = 4, p = 0.00$).

3.2 Granularity of Analysis

In information systems, time is mainly represented by time points and time intervals. Time describes the moment at which learning behaviors occur, while time interval describes the length during which the behaviors last. They are used to present a certain chronological order, cycle characteristics, and time association rules. Since the current analysis unit is the temporal data, time-related information of interest is abstracted. Each student has plenty of but usually intermittent interactions with systems. In order to summarize the statistic distributions, this study focuses on the total time during which a player is always in the playing state. The video-viewing time is an absolute measure representing the length of content students learn, while we also consider a relative measure called the viewing completion ratio, which is the proportion of video-viewing time with respect to the total video length and represents the progress of content consumption.

3.3 Preliminary observations

The most active period for students watching videos is from the November of the second half of the year to the early January of the following year. The effective learning days of the week are workdays, and ineffective learning days are weekends. The study period of the day is mainly from 9 am to 6 pm. During mealtime and other breaks such as the evenings, students rarely watch videos. Students who use mobile devices have different temporal patterns.

For each course, the cumulative playing time of most students is less than 1 minute. This shows that this part of students is not advanced students[17]. Their behavioral pattern can be attributed to “zapping style” according to[18]. There are also some students whose cumulative playing time exceeds the total length of the video and the corresponding learning completion rate is greater than 1, which indicates that these students have played the complete video from the beginning to the end or watched specific segments of the video repeatedly. In other words, their watching pattern can be attributed to repetitive style[18].

As the length of time increases, the probability density first decreases rapidly to a specific value, then reaches a peak at a faster rate and produce a thick tail. For each course, the peak of the probability density curve is relatively close in time and has a similar co-increasing or co-decreasing trend. This shows that the students' learning of the courses shares a similar distribution pattern.

The average of viewing time and the viewing completion rate are both close and low. The average cumulative viewing time for each course is about 13 minutes, and the average completion rate of video viewing is about 40%. These two values reflect the phenomena reported by most MOOC studies: high dropout rates and low resource utilization. It also shows that, compared to non-educational videos, educational videos have specific non-linear viewing patterns and a clear cognitive search intent[19].

4. RESULTS

4.1 Skewness

We perform a Kolmogorov-Smirnov normality test on the selected 7 courses. The established null hypothesis is that the viewing time or viewing completion rate conforms to a distribution of a specific normal shape. At 95% confidence and 0.05 significance level, we calculated two-tailed p-values for the two indicators which are showing equivalence to 0.000. In addition, we calculate the D statistic, which tells the maximum distance of the cumulative distribution function between the data distribution and the fitted normal distribution. More intuitively, it quantifies the magnitude of the difference between two distributions. For each course sample, the D value is large.

Besides, we observed the kurtosis and skewness coefficients. We can find that the probability density curve has a sharp peak and right-skewed shape. The right skewed distribution has a property that the mean value in the horizontal direction is greater than the median and mode[20], and the absolute value of most skewness is greater than 1.96 times its standard error, which indicates that the skewed distribution and the symmetrical distribution have statistics significance. Like the average viewing time, the distribution of viewing completion rates is all right-biased except MM, and the relationship between the median and the mean satisfies the corresponding skewed properties.

Because of data skewness, it is not appropriate to use standard methods that are based on normal distribution assumption. There are generally two methods for processing skewed data. The first method based on fitting a series of models has been implemented in[21], and the current study will try the second approach to obtain more rich features through data transformation.

4.2 Data Transformation

Intuitively, the right-biased distribution of the data causes the probability density of the long tail to change relatively slowly. This means that once students exceed the average viewing time threshold, they will have a higher proportion to invest more time in

courses. Conversely, if students' learning time does not reach this threshold, they will have a higher percentage of withdrawal from the course. Indeed, it is observable that many students fall into the second category. In order to make the curve more symmetrical, we compress the spacing of the data. This requires that the spacing of the long-tail portion is compressed faster, while the short-tail portion is compressed more slowly. After trying a lot of models empirically, we found that quadratic square root of the original data can make the data distribution basically symmetrical, and its effect is better than other available methods, such as natural logarithm. We further use the local quadratic regression on the transformed data to smooth the data. Let the range of the temporal variable X be D . For each sample $x_0 \in D$, we choose a neighborhood of x_0 . Fit the dependent variable corresponding to the observations of the temporal variables that fall within this neighborhood using the weighted least squares method. The value of the curve at x_0 is an estimate of the regression function. The results are showing in Figure 1.

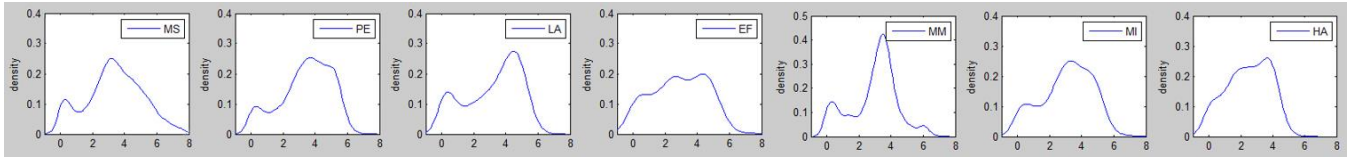


Figure 1. Curve smoothing with respect to video viewing time

The transformed time series does not satisfy any single type of distribution but presents the characteristics of a multimodal distribution. In addition to the main peak in the middle of the curve, at least one small peak may appear near the beginning or the end of the curve. In order to evaluate the goodness of fit, we use maximum likelihood estimation to fit the set of curves. Note that the current curve has a more pronounced symmetrical property near each peak than the original curve, which inspired us to try a gaussian mixture model. Suppose a curve approximates a k , $k \geq 2$ normal mixed distribution, the probability density of the curve can be expressed as:

$$p(x) = \sum_{i=1}^k \lambda_i p_i(x), \quad (1)$$

where λ_i is the weight of the i 'th gaussian distribution, satisfying $\sum_{i=1}^k \lambda_i = 1$, and $p_i(x)$ is probability density function of the i 'th gaussian distribution, satisfying $p_i(x) \sim (u_i, \sigma_i^2)$. To prevent overfitting due to empirical selection, we consider only the simplest $k = 2$ here, and compare its goodness of fit with a single normal distribution to choose the right fitting model.

Evaluation metrics include root mean square error (RMSE), adjusted R^2 , and Akaike's Information Criterion (AIC). They are showing in Table 1 that for all courses the binormal distribution is better than the single normal distribution. Numerically, it always has $RMSE_{k=2} < RMSE_{k=1}$, $R^2_{k=2} > R^2_{k=1}$, and $AIC_{k=2} < AIC_{k=1}$. We can also find that in the two models, the biggest improvement is LA, which means that its binormal distribution feature is more significant. While for MS with little improvement, both single and binormal distributions can be used for fitting. This may depend on the course setting. MS is a campus-wide elective course, which has

large number of samples and thus shows more characteristics of normal distribution according to the central limit theorem[22]. Additionally, EF, MI, and HA courses have smaller AIC values, indicating that they are suitable for a binormal distribution. In order to quantitatively evaluate the improved effect size, we focus on the AIC indicator considering that it imposes more stringent penalties on the complexity of the model compared to other indicators, so that the model we choose not only has the minimum parameters but also prevents overfitting[23, 24]. The effect size can be calculated as the improvement ratio of the AIC value while using the binormal distribution versus the single normal distribution. Results show that the most improved course when using binormal distribution fitting is HA, followed by MI, and the least improvement is MS.

The statistical characteristics of the bimodal distribution indicate that there are different dropout and retention patterns for students in all courses; courses with greater differences in the goodness of fit between single and binormal distributions, say LA and MM, show higher retention rates; instead, courses with smaller differences in goodness of fit between single and binormal

distributions such as MS show a more prominent dropout pattern, which should be given sufficient attentions by course organizers and educators.

Table 1. Evaluation of distributions.

	Single normal distribution			Binormal distribution		
	RMSE	Adj. R^2	AIC	RMSE	Adj. R^2	AIC
MS	2.297	0.912	223.8	2.236	0.917	172.9
PE	2.973	0.890	223.9	1.629	0.967	109.6
LA	4.948	0.652	325.8	1.705	0.959	118.7
EF	2.142	0.922	158.3	1.097	0.980	30.5
MM	5.461	0.780	345.6	2.233	0.963	172.7
MI	2.542	0.916	192.6	1.029	0.986	17.7
HA	2.775	0.917	208.7	0.92	0.991	-4.6

Repeating the above experimental process, we find that the distribution of the viewing completion rate is more complicated than that of the viewing time. If we use k mixed distribution for fitting, usually the goodness of fitting can be obtained when $k > 3$. Since the discussion of $k > 3$ is too complicated, we will deal with it by generalizing the model for arbitrary k values in Section 5. In order to reflect more details of the student learning process, we borrow the concept of temporal structure. We argue that it reflects the change of students' time investment when watching the courses, which is helpful for further analysis of students' preferences for in-

course parts. It should be noted that viewing time is different from time investment. The former is a static quantity that measures how much time is invested, and the latter is a dynamic quantity that measures the difference in structure of time investment.

4.3 Variation of Temporal Structure

In order to quantitatively describe the difference in the temporal structure of the viewing sequences, we use the coefficient of variation termed CV, which can be calculated by dividing the standard deviation by the mean.

$$CV = \sigma_x / \bar{x}, \quad (2)$$

where σ_x is the standard deviation and \bar{x} is the mean. The results are shown in the third column of Table 2. The numbers outside the brackets indicate the coefficient of variation of the original data, and the numbers inside the brackets indicate the transformed coefficient of variation.

Given that the amount of student watching is positively proportional to the time the student spends on the courses, we can evaluate the temporal structure of the video viewing sequence to reflect the rationality of the time allocation when students watches the courses. In this regard, Gini coefficient is a suitable indicator, which is shown in columns 6 and 7. It can be learned that the Gini coefficient with respect to viewing time (G-VT) and the Gini coefficient with respect to viewing completion rate (G-VCR) in the same course are relatively close. The two courses with the largest Gini coefficients are EF and MI, and the smallest are LA in G-VT and MM in G-VCR. This result shows that the time structure of the student's consumption ratio of EF and MI is slightly less reasonable than other courses.

Table 2. Measures of temporal structure

	CV	G-VT	G-VCR	ρ
MS	0.744(0.428)	0.399(0.234)	0.392(0.228)	0.999
PE	0.709(0.323)	0.391(0.237)	0.398(0.239)	1.000
LA	0.688(0.400)	0.377(0.223)	0.392(0.234)	1.000
EF	0.770(0.451)	0.410(0.252)	0.405(0.253)	0.990
MM	0.750(0.433)	0.400(0.244)	0.369(0.232)	0.993
MI	0.772(0.441)	0.412(0.246)	0.406(0.234)	0.992
HA	0.780(0.423)	0.397(0.234)	0.388(0.224)	0.948

It is worth noting that CV and Gini characterize the difference in the temporal structure with respect to viewing time and the completion rate of learning, but they show amazing consistency in values. According to literature[25], The Gini coefficient can be approximated as:

$$G = \frac{1}{\sqrt{3}} \frac{\sigma_y}{\bar{y}} \rho(y, r), \quad (3)$$

where σ_y represents the standard deviation, \bar{y} represents the mean, and $\rho(y, r)$ is the correlation coefficient between the student's cumulative viewing and his rank in the population. Both

the CV and Gini calculation formulas have the same component, i.e. standard deviation and mean. There is a strong positive correlation between the CV of the transformed data and the G-VT of the original data obtained by the spearman rank correlation test at a significance level of 0.01(coef. = 0.929, p = 0.003); and there is a positive correlation between the CV of the transformed data and the transformed G-VT at a significance level of 0.05(coef. = 0.683, p = 0.033). Bringing the mean and standard deviation of Table 2 into the formula, we obtain the ρ values.

5. Course Similarity

5.1 Metric

In order to make the model scalable, we assume that other courses can also be fitted with the gaussian mixture model by choosing the appropriate k value.

For each course, we run GMM clustering algorithm and obtain parameters of GMM. The log-likelihood function can be written as:

$$l(\pi, \mu, \sigma) = \sum_{i=1}^N \log \left(\sum_{k=1}^K \pi_k P(x_i; \mu_k, \sigma_k) \right). \quad (4)$$

K-means algorithm is used to initialize model parameters and EM algorithm is used to optimize the parameters.

Given two course samples C^i and C^j , we model them as gaussian mixture model $\Omega^i = \{\phi_1^i, \phi_2^i, \dots, \phi_{K_1}^i\}$ and $\Omega^j = \{\phi_1^j, \phi_2^j, \dots, \phi_{K_2}^j\}$

where K_1 and K_2 are the number of components of Ω^i and Ω^j respectively. Then, the average similarity of the two course distributions[26] can be computed by:

$$S(\Omega^i, \Omega^j) = 1 - \frac{1}{K_1 K_2} \sum_{h=1}^{K_1} \sum_{l=1}^{K_2} d(\phi_h^i, \phi_l^j), \quad (5)$$

$$d(\phi_m^i, \phi_n^j) = \frac{1}{8} (\mu_m^i - \mu_n^j)^T \left(\frac{\Sigma_m^i + \Sigma_n^j}{2} \right)^{-1} (\mu_m^i - \mu_n^j) + \frac{1}{2} \ln \frac{\left| \frac{\Sigma_m^i + \Sigma_n^j}{2} \right|}{\sqrt{|\Sigma_m^i| |\Sigma_n^j|}}, \quad (6)$$

where $d(\phi_m^i, \phi_n^j)$ is the Bhattacharyya distance that measures the pair-wise similarity of multivariate normal distributions. The bigger of the S value, the similar of the course samples.

5.2 Discussion

The distribution-based course similarity can be applied to personalized course recommendation that addresses the information overload issue by customizing the learning content for students[27]. In previous studies, teachers describe the attributes of courses by analyzing their content, or pre-define corresponding learning goals as the extent to which students would acquire knowledge and skills[28]. The obvious limitation is the lack of a dynamic description of the learning process. Existing course similarity calculation are mainly based on the traditional text mining approaches with a vector space model been constructed according to the knowledge points that each course contains[29]. They mainly suffer from not considering the real-time temporal access patterns towards courses. The courses in the same cluster summarize students' similar learning patterns, which is helpful for assessing the learning process of students.

6. REFERENCES

- [1] Z. A. Pardos, "Big data in education and the models that love them," *Current opinion in behavioral sciences*, vol. 18, pp. 107-113, 2017.
- [2] R. S. Baker, Y. Wang, L. Paquette, V. Aleven, O. Popescu, J. Sewall, et al., "Educational data mining: A MOOC experience," *Data Mining And Learning Analytics: Applications in Educational Research*, pp. 55-66, 2016.
- [3] L. C. Liñán and Á. A. J. Pérez, "Educational Data Mining and Learning Analytics: differences, similarities, and time evolution," *International Journal of Educational Technology in Higher Education*, vol. 12, pp. 98-112, 2015.
- [4] N. Buniyamin, U. bin Mat, and P. M. Arshad, "Educational data mining for prediction and classification of engineering students achievement," in *2015 IEEE 7th International Conference on Engineering Education (ICEED)*, 2015, pp. 49-53.
- [5] N. Shahadat and B. Pal, "An empirical analysis of attribute skewness over class imbalance on Probabilistic Neural Network and Naïve Bayes classifier," in *2015 International Conference on Computer and Information Engineering (ICCIE)*, 2015, pp. 150-153.
- [6] H. Guo and H. L. Viktor, "Learning from imbalanced data sets with boosting and data generation: the databoost-im approach," *ACM Sigkdd Explorations Newsletter*, vol. 6, pp. 30-39, 2004.
- [7] W. Xing, X. Chen, J. Stein, and M. Marcinkowski, "Temporal predication of dropouts in MOOCs: Reaching the low hanging fruit through stacking generalization," *Computers in Human Behavior*, vol. 58, pp. 119-129, 2016/05/01/ 2016.
- [8] T. Xie, Q. Zheng, and W. Zhang, "Mining temporal characteristics of behaviors from interval events in e-learning," *Information Sciences*, vol. 447, pp. 169-185, 2018.
- [9] I. Molenaar and S. Järvelä, "Sequential and temporal characteristics of self and socially regulated learning," *Metacognition and Learning*, vol. 9, pp. 75-85, 2014.
- [10] H. C. Ratz, "Extreme values from skewed distributions [mathematics education]," *IEEE Transactions on Education*, vol. 43, pp. 400-402, 2000.
- [11] M. Blanca, R. Alarcón, J. Arnau, R. Bono, and R. Bendayan, "Non-normal data: Is ANOVA still a valid option?," *Psicothema*, vol. 29, pp. 552-557, 2017.
- [12] J. W. Osborne, "Improving your data transformations: Applying the Box-Cox transformation," *Practical Assessment, Research & Evaluation*, vol. 15, pp. 1-9, 2010.
- [13] A. Belussi, S. Migliorini, and A. Eldawy, "Detecting skewness of big spatial data in SpatialHadoop," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2018, pp. 432-435.
- [14] X. Wang, K. Smith, and R. Hyndman, "Characteristic-based clustering for time series data," *Data mining and knowledge Discovery*, vol. 13, pp. 335-364, 2006.
- [15] Q. Cai, L. Chen, and J. Sun, "Piecewise statistic approximation based similarity measure for time series," *Knowledge-Based Systems*, vol. 85, pp. 181-195, 2015.
- [16] I. Ozken, D. Eroglu, T. Stemler, N. Marwan, G. B. Bagci, and J. Kurths, "Transformation-cost time-series method for analyzing irregularly sampled data," *Physical Review E*, vol. 91, p. 062911, 2015.
- [17] C. G. Brinton, S. Buccapatnam, M. Chiang, and H. Poor, "Mining MOOC Clickstreams: On the Relationship Between Learner Behavior and Performance," *arXiv preprint arXiv:1503.06489*, 2015.
- [18] J. De Boer, P. A. Kommers, and B. De Brock, "Using learning styles and viewing styles in streaming video," *Computers & Education*, vol. 56, pp. 727-735, 2011.
- [19] O. A. Acar and J. van den Ende, "Knowledge Distance, Cognitive-Search Processes, and Creativity: The Making of Winning Solutions in Science Contests," *Psychological science*, vol. 27, pp. 692-699, 2016/05// 2016.
- [20] Y. Chen, B. Zhang, Y. Liu, and W. Zhu, "Measurement and modeling of video watching time in a large-scale internet video-on-demand system," *IEEE Transactions on Multimedia*, vol. 15, pp. 2087-2098, 2013.
- [21] T. Xie, Q. Zheng, W. Zhang, and H. Qu, "Modeling and predicting the active video-viewing time in a large-scale E-learning system," *IEEE Access*, vol. 5, pp. 11490-11504, 2017.
- [22] W. Hoeffding and H. Robbins, "The central limit theorem for dependent random variables," pp. 773-780, 1948/09 1948.
- [23] H. Bozdogan, "Model selection and Akaike's Information Criterion (AIC): The general theory and its analytical extensions," *Psychometrika*, vol. 52, pp. 345-370, 1987// 1987.
- [24] K. P. Burnham and D. R. Anderson, "Multimodel inference - understanding AIC and BIC in model selection," *Sociological Methods & Research*, vol. 33, pp. 261-304, Nov 2004.
- [25] B. Milanovic, "A simple way to calculate the Gini coefficient, and some implications," *Economics Letters*, vol. 56, pp. 45-49, 1997.
- [26] Z. Yu, X. Zhu, H.-S. Wong, J. You, J. Zhang, and G. Han, "Distribution-based cluster structure selection," *IEEE transactions on cybernetics*, vol. 47, pp. 3554-3567, 2016.
- [27] A. Parameswaran, P. Venetis, and H. Garcia-Molina, "Recommendation systems with complex constraints: A course recommendation perspective," *ACM Transactions on Information Systems (TOIS)*, vol. 29, p. 20, 2011.
- [28] A. Pawar and V. Mago, "Similarity between learning outcomes from course objectives using semantic analysis, blooms taxonomy and corpus statistics," *arXiv preprint arXiv:1804.06333*, 2018.
- [29] B. Zhao and X. Li, "Course Similarity Calculation Using Efficient Manifold Ranking," in *International Conference on Knowledge Science, Engineering and Management*, 2015, pp. 421-432.

Semi-supervised Learning Method for Adjusting Biased Item Difficulty Estimates Caused by Nonignorable Missingness under 2PL-IRT Model ^{*}

Kang Xue
NWEA & University of Georgia
kang.xue@nwea.org,
kangxue@uga.edu

Anne Corinne
Huggins-Manley
University of Florida
amanley@coe.ufl.edu

Walter Leite
University of Florida
Walter.Leite@coe.ufl.edu

ABSTRACT

In data collected from virtual learning environments (VLEs), item response theory (IRT) models can be used to guide the ongoing measurement of student ability. However, such applications of IRT rely on unbiased item parameter estimates associated with test items in the VLE. Without formal piloting of the items, one can expect a large amount of non-ignorable missing data in the VLE log file data, and this is expected to negatively impact IRT item parameter estimation accuracy, which then negatively impacts any future ability estimates utilized in the VLE. In the psychometric literature, methods for handling missing data are mostly centered around conditions in which the data and the amount of missing data are not as large as those that come from VLEs. In this paper, we introduce a semi-supervised learning method to deal with a large proportion of missingness contained in VLE data from which one needs to obtain unbiased item parameter estimates. The proposed framework showed its potential for obtaining unbiased item parameter estimates that can then be fixed in the VLE in order to obtain ongoing ability estimates for operational purposes.

Keywords

virtual learning environment, semi-supervised learning, item response theory, missing data

1. INTRODUCTION

In contrast to physical learning environments such as classrooms, a virtual learning environment (VLE) refers to a system that delivers learning materials to students in a digital space. Item response theory (IRT) [3] refers to a family

of mathematical models that attempt to explain the relationship between latent traits (unobservable skills or knowledge) and their manifestations (i.e. observed outcomes, responses or performance) using different statistic functions (e.g. Rasch Model, 2PL-IRT, multidimensional IRT). To estimate the item parameters for further personal adaptive learning (e.g., providing appropriate item which matches student's ability could encourage student to complete it), IRT models are widely used to determine the psychometric properties of items through analyzing students' responses in VLE [9].

How to reduce the impact of missing values on item parameter estimation of IRT models is a very common issue for data analysis and attracts lots of research attention. Generally, missing values could be categorized to 4 classes: structurally missing data, missing completely at random (MCAR), missing at random (MAR) and missing not at random (i.e. non-ignorable missing values) [12]. In contrast to other types of missing values, nonignorable missing values in assessment are more complicated because they are usually caused by latent factors to be measured by IRT models. For assessment data, researchers has proposed different model-based approaches to reduce the impacts from nonignorable missing values [10]. One model-based approach, the latent approach, includes missing tendency via a latent missing propensity that is accounted for in a multidimensional IRT model [4]; another model-based approach, the manifest approach, includes missing tendency by modeling a manifest missing variable that is accounted for in a unidimensional missingness propensity [11].

However, in contrast to assessment, the data collected in VLE often contain large proportion of missingness when students are allowed to skip questions in some online courses. It makes that the missing data in VLEs are caused by a variety of cognitive and motivational factors (e.g., excess challenge, lack of challenge or lack of time). The model-based approaches are not suitable to deal with such kinds of missingness in the data collected from VLE, because determining the latent missing propensity will be very complicated for drawing inferences to model the joint distribution of the missingness and the item responses [6].

The technological changes across learning, instruction and assessment start to bring machine learning techniques into psychometrics because machine learning algorithms have the

^{*}The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305C160004 to the University of Florida. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education.

capability to analyzing complex and high-dimensional data. Applying data mining and machine learning techniques to VLE data is a mechanism to improve research in technology-enhanced educational environments [1, 8]. For example, IRT psychometric models are usually based upon logistic regression techniques which are used to be popular in solving classification problem in machine learning [7]. As a sub-field of machine learning, the primary goal of deep learning is to extract the latent variables from the input distribution using artificial neural networks (ANNs) which is a computational system inspired by biological neural networks [5]. In educational research area, deep learning has been applied for different tasks, such as automatic item generation (AIG) [14], automated scoring [13], and item characteristics prediction [17].

Inspired by the research using deep learning and semi-supervised learning techniques for cognitive diagnostic classification [15], we proposed a semi-supervised deep learning framework to reduce the impact on item parameter estimation caused by nonignorable missing values when applying two-parameter IRT (2PL-IRT) model to the data collected in VLE. The research in this paper consists of two parts: (1) exploring the real data collected within a statewide-used VLE to test if the missingness was caused by student ability and item difficulty which were measured in 2PL-IRT; and (2) proposing a semi-supervised learning method using deep learning techniques to adjust the bias in estimation caused by missingness. In the following part of this paper, we will firstly introduce the operational data exploration on the data collected within a VLE; then the semi-supervised learning method will be described in detail; the simulated study shows the performance of the proposed framework in dealing with nonignorable missingness; lastly, we will conclude the findings and limits in this framework and discuss some potential future research.

2. OPERATIONAL DATA EXPLORATION

The data collected in this research were students' responses to the "Algebra I" items within a statewide-used VLE system. The dataset contains 10 algebra domains, and we treated each domain as having its own ability to measure. The number of items ranged from 41 to 89 across domains. The total number of students was 63,625. Since students were allowed to skip items in the learning environment when they responded to the items which were selected by the system randomly, the responses to each item contained large amount of missing values. The proportion of missingness for each item is between 55% to 75%. Generally, the response patterns of students could be classified into 3 categories: 1) skipped the domain (i.e., no responses to any test items within the domain), 2) completed the domain (i.e., responded to all test items within the domain), 3) mixed response (i.e., responded to some items within the domain).

To test if the missingness was related to the item and person parameters in the 2PL-IRT model, a hierarchical logistic regression (Figure 1) was conducted for each domain individually. The hierarchical logistic regression was consisted of (1) **skipping domain test** was to test if skipping a domain related to the students' ability; (2) **completing domain test** was to test if completing a domain related to the students' ability; (3) **mixed response test** was to test if student

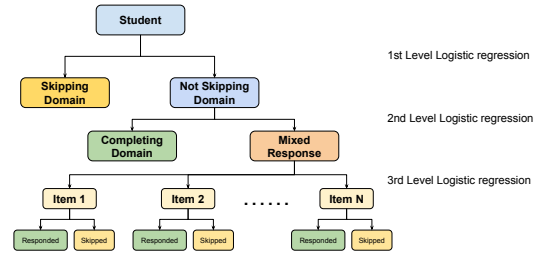


Figure 1: The diagram of the hierarchical logistic regression.

skipping an item related to the item difficulty and students ability. As an area of mathematics, there is high correlation between the math skills and algebra skills. Thus, we used the pretest mathematical scores on the state standardized test, S , as student's true ability for the data exploration. To evaluate the relationship between ability and skipping a domain, all the students' responses were classified to two groups: students skipped the domain and students didn't skip the domain. The second group contained students completed the domain and students with mixed responses. Then the logistic regression test was conducted for each school district individually as following:

$$\text{logit}(\text{skipping domain}) = \beta_{0,ij} + \beta_{1,ij}S \quad (1)$$

where j indicates the j th educational district and i refers to the i th domain. After fitting the models, we found that for most school districts and most students, $\beta_{1,ij}$ were significant negative. We can conclude that students with high ability level had a lower probability to skip a domain, and students with low ability level had higher probability to skip a domain.

After doing skipping domain test, in the completing domain test, the dataset only contained students who didn't skip the domain. The dataset was divided to two groups: students completed domain and students with mixed response. The logistic regression test was conducted for each school district individually as following:

$$\text{logit}(\text{completing domain}) = \beta_{0,ij} + \beta_{1,ij}S \quad (2)$$

where j indicates the j th educational district and i refers to the i th domain. In contrast to the observation of "skipping domain", it was not reasonable to reach a consistent conclusion about the relationship between the ability and completing domain.

In the last subtest, two factors, students' ability and item difficulty, were assumed to impact the probability that a student responded to an item. We chose the observed incorrect response rate of the item, D_k , to indicate the item difficulty. The logistic regression was as following:

$$\text{logit}(\text{skipping } k\text{th item}) = \beta_{0,ik} + \beta_{1,ik}S + \beta_{2,ik}D_k \quad (3)$$

where i is the i th domain and k indicates the k th item. The logistic regression test showed that students with lower ability level had higher probability to skip an item shown to them; and student had a higher probability to skip item with higher difficulties. From the data exploration, we could conclude that the missing values in the data collected contained

nonignorable missingness because they were caused the factors have relationship with the latent variables measured in the 2PL-IRT model.

3. SEMI-SUPERVISED DEEP LEARNING-BASED BIAS ADJUSTMENT

Intuitively, there was no missing value in the response from the anchor students, who completed all items in a domain. However, directly applying 2PL-IRT model to the anchor students would impact a parameter invariance because from the data exploration also showed there existed difference between the sub-population of anchor students and whole population. To adjust the biased ability estimates and item parameters estimates through directly applying 2PL-IRT model to the anchor students, we proposed a semi-supervised deep learning-based bias adjustment procedure which consisted of the unbiased ability estimation through a semi-supervised deep learning architecture, and the item parameter adjustment methods.

3.1 Semi-supervised deep learning architecture

The thinking of semi-supervised learning was used to improve the robustness of binary latent person variables (e.g. attribute mastery status) estimation [15]. In this research, because the latent person variables measured in 2PL-IRT model were continuous, the semi-supervised learning techniques were conducted based on the following two assumptions:

1. Given the unbiased latent trait Θ for each student, the biased estimation $\hat{\theta}$ directly using 2PL-IRT could be represented through a function: $\hat{\theta} = \phi(\Theta)$;
2. The unbiased latent trait Θ could maximize the likelihood function $P(X = 1; \Theta) = L(\Theta)$ which indicates the relationship between latent trait Θ and item response pattern $\mathbf{X} = \{x\}$;

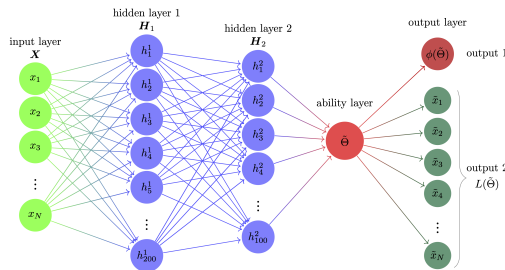


Figure 2: The diagram of the proposed semi-supervised deep learning architecture for unbiased ability estimation. In this framework, a deep learning architecture with 3 hidden layers was used to convert the observed response patterns to the unbiased ability. To train the deep learning architecture, the distance between two outputs of the DFN and two targets was minimized.

Regarding to these assumptions, the goals of the proposed semi-supervised deep learning structure to extract the unbiased latent trait Θ from the anchor students response data and approximate the function $\phi(\Theta)$ which indicated the relationship between unbiased latent trait Θ and biased estimation $\hat{\theta}$ and $L(\Theta)$ which indicates the relationship between

latent trait Θ and item response pattern $\mathbf{X} = \{x\}$. From Figure 2, there were three hidden layers between the input layer and the latent trait layer. The number of hidden layers were set based on the previous research of using deep learning method for cognitive diagnostic models [16, 2]. To bring the nonlinearity to the DFN, Rectified Linear Units (ReLU) was chosen as the activation function. The unbiased latent trait $\hat{\Theta}$ extracted using the DFN could be represented as: $\hat{\Theta} = \Phi(\mathbf{X}; \omega)$. ω were the connection weights in the DFN. The parameters of DFN, ω , were estimated by minimizing the following weighted cost function:

$$\omega = \arg \min(w_1 MSE(\hat{\theta}, \phi(\hat{\Theta})) + w_2 H(\hat{\mathbf{X}}, \mathbf{X})) \quad (4)$$

where $\hat{\theta}$ is the biased students' ability estimation directly fitting 2PL-IRT model to the anchor students' responses; $\mathbf{X} = \{x\}$ is the observed response patterns of the anchor students. In the weighted cost function, we used two kinds of error functions corresponding to two outputs respectively: the mean square error (MSE) was used to calculate the difference between continuous variables $\hat{\theta}$ and $\phi(\hat{\Theta})$; the cross-entropy (H) was used to calculate the difference between binary variables \mathbf{X} and $\hat{\mathbf{X}}$. The two hyperparameters, w_1 and w_2 , were determined using the elbow method in validation test.

3.2 Two item parameter adjustment methods

After obtaining the parameter estimation through the training procedure, the DFN converted observed response pattern \mathbf{X} to unbiased ability estimation $\hat{\Theta}$. To reduce the biases contained in the item difficulty, two kinds of adjustment methods, item equating adjustment (IEA) and bootstrapping adjustment (BA), were proposed using the unbiased ability estimation $\hat{\Theta}$.

IEA was inspired by the common group equating design in IRT. In IEA, the ability distribution of anchor students was the frame of reference. Then the biased item difficulty estimates were placed onto unbiased item difficulty via $\tilde{b}_j = \hat{b}_j - (\hat{\Theta} - \bar{\hat{\Theta}})$. $\bar{\hat{\theta}}$ and $\bar{\hat{\Theta}}$ are the average of biased ability estimates and unbiased ability estimates respectively, \hat{b}_j is the biased item difficulty estimates for j th item, and \tilde{b}_j is the adjusted item difficulty estimates. IEA only reduced the biases contained in the item difficulty estimates because it held an assumption that the item discrimination estimates were not biased.

In contrast to IEA, BA was proposed to reduce the biases contained in both item difficulty and item discrimination parameters using bootstrapping in statistics. There were 4 steps contained in BA method:

1. Randomly sampled from the anchor students based on the unbiased ability estimates $\hat{\Theta}$ to make the ability distribution of the new sample set is standard normal distribution and the sample size was same as the original anchor students;
2. Apply 2PL-IRT to the new sample set and estimate the item difficulty parameters and item discriminating parameters;
3. Repeated step 1 and step 2 K times, a group of estimates of difficulty and discriminating of j th item could be obtained $\{\tilde{a}_{j,k}, \tilde{b}_{j,k}\}$, where $k = 1, K$;

Table 1: Comparison of the distribution of ability estimates between directly 2PL-IRT model fitting ($\hat{\theta}$) and the proposed semi-supervised deep learning architecture ($\tilde{\theta}$).

Domains	True $\Theta(\sigma)$	$\hat{\theta}(\sigma)$	$\tilde{\theta}(\sigma)$
1	0.090 (0.93)	-0.001 (0.99)	0.095 (0.90)
2	0.169 (0.85)	0.000 (0.98)	0.157 (0.82)
3	0.203 (0.83)	0.000 (1.01)	0.198 (0.85)
4	0.152 (0.88)	-0.001 (0.99)	0.160 (0.81)
5	0.178 (0.87)	-0.001 (1.00)	0.180 (0.88)
6	0.228 (0.75)	-0.001 (0.99)	0.232 (0.73)
7	0.168 (0.85)	-0.001 (1.01)	0.171 (0.83)
8	0.218 (0.79)	-0.000 (1.00)	0.207 (0.80)
9	0.241 (0.77)	-0.000 (1.00)	0.241 (0.79)
9	0.312 (0.72)	-0.000 (0.98)	0.320 (0.69)

4. Then the estimate of item discrimination equaled to $\frac{1}{K} \sum_1^K \tilde{a}_{j,k}$, and the estimate of item difficulty equaled to $\frac{1}{K} \sum_1^K \tilde{b}_{j,k}$.

The BA method relies on less constraint and could reduce the biases contained in both item discrimination and difficulty estimates. The BA has the potential for applying on more complicated IRT models, such as 3PL-IRT.

4. SIMULATED STUDY

The proposed methods were tested through a simulation study under 2PL-IRT model. In the simulated study, we used “mirt” package in R to conduct data simulation and IRT model fitting and used “Tensorflow” toolbox in python to achieve the unbiased ability estimates through the semi-supervised deep learning architecture. To create data under 2PL-IRT, the known pretest mathematical ability were used as the students’ ability, and the biased item parameters obtained through directly applying 2PL-IRT to the anchor students were used as item parameters. The fitted functions 1, 2, and 3 in data exploration were used to predict the students’ response patterns (e.g., skipping domain, completing domain, mixed response). We selected the response of anchor students who completed all items in a domain as the input of our proposed method.

First, we applied the 2PL-IRT model directly to the simulated anchor students’ responses for each domain to estimate the item parameters and students’ ability. Then, the proposed semi-supervised deep learning architecture was applied using the simulated anchor students’ responses as input and using the anchor students’ ability estimates and their response patterns as two targets. By minimizing the weighted cost function in Equation 4, the unbiased ability of anchor students was estimated. The validating test was conducted in the training procedure to avoid over-fitting and determine the two hyperparameters w_1 and w_2 in Equation 4. Table 4 compares the distribution of ability estimates between directly 2PL-IRT model fitting and the proposed semi-supervised deep learning architecture.

Using the estimation of the anchor students’ ability through the semi-supervised deep learning architecture, the two proposed adjustment methods, IEA and BA, were conducted to reduce the biases contained in the item difficulty parameters. We chose two criteria, rooted mean squared er-

ror (RMSE) and variance, to evaluate the bias adjustment methods. RMSE indicates the distance between item difficulty estimates and true item difficulty parameters, and the variance indicates the consistency of the estimates from different methods. From Figure 3, in contrast to the directly 2PL-IRT model fitting, both IEA and BA achieved much less RMSE for each domain. For variance, since the IEA adjusted the difficulty estimates based on a parallel shift of the ability distribution, the variance of IEA and directly 2PL-IRT results were the same. However, the BA method obtained more consistent estimates because bootstrapping in BA created standard normal distributed samples which matched the assumption of original IRT estimation. From the experimental results, both IEA and BA had the ability to adjust the biases contained in the estimates of item difficulty using directly 2PL-IRT model fitting. Compared with IEA which only reduce the biases of item difficulty parameters, BA method had the potential to reduce the biases contained in the item parameters for different IRT models.



Figure 3: Comparison of the item difficulty estimates among direct applying 2PL-IRT model fitting, item equating adjustment (IEA) and bootstrapping adjustment (BA).

5. CONCLUSION

Nonignorable missingness impacts applying psychometric models to the data collected in VLE. To reduce the impacts of nonignorable missingness, this research explored a statewide-used VLE data to test the hypothesis that the missing values were non-ignorable missingness and related to the factors that 2PL-IRT model measures. The data exploration showed that the non-ignorable missingness would impact the parameter estimation of 2PL-IRT without pre data analysis. To adjust the biased item difficulty parameter estimates caused by the non-ignorable missingness, a semi-supervised learning framework was designed. In the framework, the idea of semi-supervised learning was first time used in IRT area to improve the robustness of latent trait estimation. To convert the observed response pattern to the continuous latent trait and approximate some continuous functions which were hard to specify mathematically, deep learning techniques were also introduced. The combination of semi-supervised deep learning and IRT model improved both accuracy and robustness of the parameter estimation for IRT on noisy data with weak constraint. The experimental results showed that the proposed framework adjust the biases contained in both students’ ability estimation and item parameter estimation for 2PL-IRT model.

6. REFERENCES

- [1] M. Bienkowski, M. Feng, B. Means, et al. Enhancing teaching and learning through educational data mining and learning analytics: An issue brief. *US Department of Education, Office of Educational Technology*, 1:1–57, 2012.
- [2] Y. Cui, Q. Guo, and M. Cutumisu. A neural network approach to estimate student skill mastery in cognitive diagnostic assessments. 2017.
- [3] S. E. Embretson and S. P. Reise. *Item response theory*. Psychology Press, 2013.
- [4] R. Holman and C. A. Glas. Modelling non-ignorable missing-data mechanisms with item response theory models. *British Journal of Mathematical and Statistical Psychology*, 58(1):1–17, 2005.
- [5] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [6] F. M. Lord. Maximum likelihood estimation of item response parameters when some responses are omitted. *Psychometrika*, 48(3):477–482, 1983.
- [7] F. Martínez-Plumed, R. B. Prudêncio, A. Martínez-Usó, and J. Hernández-Orallo. Making sense of item response theory in machine learning. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, pages 1140–1148. IOS Press, 2016.
- [8] B. Means and K. Anderson. Expanding evidence approaches for learning in a digital world. *Office of Educational Technology, US Department of Education*, 2013.
- [9] J. Y. Park, T. Dougherty, H. Fritz, and Z. Nagy. Lightlearn: An adaptive and occupant centered controller for lighting based on reinforcement learning. *Building and Environment*, 147:397–414, 2019.
- [10] S. Pohl, L. Gräfe, and N. Rose. Dealing with omitted and not-reached items in competence tests: Evaluating approaches accounting for missing responses in item response theory models. *Educational and Psychological Measurement*, 74(3):423–452, 2014.
- [11] N. Rose, M. Von Davier, and X. Xu. Modeling nonignorable missing data with item response theory (irt). *ETS Research Report Series*, 2010(1):i–53, 2010.
- [12] D. B. Rubin. The calculation of posterior distributions by data augmentation: Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The sir algorithm. *Journal of the American Statistical Association*, 82(398):543–546, 1987.
- [13] K. Taghipour and H. T. Ng. A neural approach to automated essay scoring. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1882–1891, 2016.
- [14] M. von Davier. Automated item generation with recurrent neural networks. *psychometrika*, 83(4):847–857, 2018.
- [15] K. Xue. Computational diagnostic classification model using deep feedforward network based semi-supervised learning. In *25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) Workshop on Deep Learning for Education*, 2019.
- [16] K. Xue, V. Yaneva, and C. Runyon. On the utility of using transfer learning to predict item characteristics. In *2020 Annual Meeting of the National Council on Measurement in Education (NCME)*, 2020.
- [17] K. Xue, V. Yaneva, C. Runyon, and P. Baldwin. Predicting the difficulty and response time of multiple choice questions using transfer learning. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2020.

An effect-size-based temporal interestingness metric for sequential pattern mining

Yingbin Zhang

University of Illinois at Urbana-Champaign

yingbin2@illinois.edu

Luc Paquette

University of Illinois at Urbana-Champaign

lpag@illinois.edu

ABSTRACT

Sequential pattern mining is a useful technique for understanding learning behavior. However, it can be challenging to select the most “interesting” patterns discovered through sequence mining. The work presented in this paper proposes an effect-size-based (ESB) method to help researchers identify temporally interesting sequential patterns. ESB is extended from the Temporal Interestingness of Patterns in Sequences (TIPS) technique [4] and distinguishes itself by 1) considering a different association direction between the sequential pattern usage and time, 2) providing a more interpretable ranking metric, and 3) providing a different ranking order for temporally interesting sequential patterns. Both ESB and TIPS are applied to interaction log data to demonstrate their differences in selecting sequential patterns.

Keywords

Sequential pattern mining, effect size, interestingness metric, learning behavior evolution.

1. INTRODUCTION

Sequential pattern mining (SPM) aims to find temporal relationships between events [1]. It is a useful tool to understand students’ learning behavior and becomes increasingly popular in the field of education [10, 18]. For example, SPM has been applied to investigate the evolution of cognitive and metacognitive behavior within a computer-based science learning environment [7], to understand students’ problem-solving behavior and to explore the associations among metacognitive monitoring, scientific inquiry skills, and task performance within game-based learning environments [4, 16].

Due to the exploratory nature of SPM, researchers need to expend considerable efforts to interpret them and obtain actionable insight for teaching and learning from the discovered sequential patterns [5]. However, the number of sequential patterns discovered through SPM may be huge, and, as such, it is inefficient and sometimes impossible to investigate these patterns one by one. To ease selecting patterns, researchers proposed interestingness metrics to rank sequential patterns or association rules [9].

There has been interest in the topic of temporal analyses of learning data [11], especially in the context of self-regulated learning [12].

Yingbin Zhang and Luc Paquette “An effect-size-based temporal interestingness metric for sequential pattern mining” In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 720 - 724

As such, patterns that vary across time may be particularly interesting because they may reveal additional information. For instance, the variation of pattern occurrences across time can be used to evaluate the effectiveness of learning support [7]. If the evolution of some sequential patterns in the group who received support from the environment is different from the group without support, the support may have effects on learners’ behavior. The evolution of sequential patterns may also provide insights into improving the learning environment. For example, if a sequential pattern beneficial for learning frequently occurs during the whole learning processes except for a particular period, what happens in this period may be interesting. Understanding events in this period may further inform designing intervention to prevent students from stopping this behavior pattern in this period.

In order to ease the selection of temporally interesting patterns, Kinnebrew, Segedy, and Biswas [5] proposed the Temporal Interestingness of Patterns in Sequences (TIPS) technique, an information gain-based approach, to rank patterns contingent on their variation across time. This research extends the TIPS technique by proposing an effect-size-based (ESB) method. ESB was applied to interaction log data of students’ using the Betty’s Brain learning environment [2] to demonstrate its relative advantages in identifying temporally interesting sequential patterns in comparison with TIPS.

1.1 The procedure of TIPS

TIPS firstly segments each student’s log file into n ordered bins (e.g., five ordered bins) with equal sizes [5]. Then, for each student, it calculates the occurrences (also known as instance values) of each frequent sequential pattern in each bin. Thirdly, it takes the occurrences of a pattern per bin per student as the feature and the bin number (e.g., 1, 2, 3, 4, 5) as the label and calculates the information gain (IG) of this pattern. IG refers to the reduction in Shannon entropy about the label from knowing the feature. Its calculation is [14]:

$$IG(L, F) = Entropy(L) - Entropy(L|F) \quad (1)$$

L refers to the label, while F refers to the feature. $Entropy(L)$ is the priori Shannon entropy about the label, while $Entropy(L|F)$ is the conditional Shannon entropy about the label given the feature. Finally, TIPS ranks all frequent sequential patterns based on their IG, and the top-ranking sequential patterns may be temporally interesting.

2. Effect-size based (ESB) temporal interestingness metric

2.1 The procedures of ESB

The ESB approach also needs the first two steps of TIPS, i.e., computing the occurrences per bin per student for each sequential pattern. However, in the next step, the ESB adopts the idea of

repeated-measures designs [13] and regards the occurrences of a pattern as a variable that is measured several times. One bin is one time. Under this framework, one-way repeated ANOVA can be conducted with the occurrences as the dependent variable and the bin number (i.e., the time) as the independent within-subject variable. Then, ESB calculates the effect size to characterize the association strength between the bin number and the occurrences of a pattern. The ESB regards the effect size as a temporal interestingness metric for sequential patterns. Given that the number of students and bins within a study is constant, the sample sizes for all frequent patterns are the same. Thus, the effect size is comparable across sequential patterns.

There are several effect size measures for ANOVA. Lakens [8] suggested using omega squared for comparisons of effects within a study. The meaning of omega squared is analogous to R squared in linear regression. It estimates the percent of variance explained by the independent variable (the bin in this case). For instance, an omega squared of 0.1 means that 10% of pattern occurrence variance can be explained by the bin number (i.e., time).

Omega squared is used for parametric repeated ANOVA. However, in practice, the distribution of sequential pattern occurrences may violate the assumptions of parametric ANOVA, such as the normality assumption and the homogeneity of variance. For example, some temporally varying sequential patterns may rarely happen at the beginning or end of learning activities. Their occurrence values have many zero in these periods, and their distributions are highly skewed. In this case, it would be better to conduct a non-parametric repeated ANOVA, such as the Friedman test. The effect size corresponding to the Friedman test is Kendall's W [17]. Its calculation is:

$$W = \frac{\chi^2}{N(k-1)} \quad (2)$$

χ^2 is the Friedman test statistic value. N is the number of subjects, and k is the number of measurements per subject. Kendall's W is interpreted similarly to omega squared and ranges from 0 (no relationship) to 1 (perfect relationship).

2.2 Differences between TIPS and ESB

2.2.1 Implicit assumptions.

The direction of the relationship between the occurrences of patterns and time is opposite in the two methods. TIPS examines the extent to which the occurrences of a pattern can distinguish different bins. In other words, TIPS implies that the occurrences of a pattern influence the bin number. In contrast, the ESB assumes that the bin number influences the occurrences of a pattern. While both approaches look at the evolution of the usage of patterns, ESB's assumption is more natural since the assumption is that the bin number is fixed, and the frequency of the pattern is what varies over time. Nevertheless, this distinction between the TIPS and ESB is conceptual and may not have a practical impact.

2.2.2 Interpretability.

The interpretability of ESB may be better than TIPS. As demonstrated above, the meaning of effect size (e.g., omega squared and Kendall's W) is straightforward. Besides, for researchers having experiences with ANOVA, they may already be more familiar with such measures of effect size. This characteristic of ESB can facilitate setting a threshold to filter patterns that may be less temporally interesting. For example, a general rule of thumb on magnitudes of Kendall's W is that W higher than 0.1 but smaller than 0.3 represents a small effect, W no less than 0.3 but less than

0.5 represents a medium effect, and W no less than 0.5 is a large effect [3]. If researchers are only interested in patterns that have at least medium variation across time, they can use 0.3 as the Kendall's W threshold to filter patterns. However, it is more challenging to decide the information gain threshold because the scale of information gains depends on contexts, such as the number of categories (i.e., the number of bins) of the label and the number of distinct values of the feature.

3. Application example

In order to demonstrate the differences of TIPS and ESB in identifying temporally interesting sequential patterns, they were applied to data from a recent study where 88 sixth-grade students learned climate change within Betty's Brain, a computer-based learning environment [2]. Students firstly received a training session on how to use Betty's Brain and used it to study climate change in the next four school days around 45 minutes per day. The action logs of students' working on Betty's Brain were analyzed. The output of TIPS and ESB were compared to investigate the relative advantages of ESB.

3.1 Betty's Brain

In Betty's Brain, students learn about scientific phenomena, such as climate change, by teaching Betty, a virtual pedagogical agent. They teach Betty by adding scientific concepts and directed causal links among the concepts on a blank page. Students can access hypermedia resource pages on relevant scientific concepts and causal relationships. Students can evaluate the causal links by asking Betty to take quizzes. By looking at Betty's correct and incorrect answers, students can identify problems in their understanding.

3.2 Data preprocessing

Firstly, irrelevant actions, such as actions initiated by the system, were removed from the raw action logs [6]. Then, actions were contextualized based on the duration and coherence. Viewing quiz results actions were labeled long vs. short, depending on whether the duration was higher than 3 seconds. Reading page actions were labeled long vs. short, depending on whether the duration was longer than 10 seconds. Long reading pages, adding, revising, and marking links were labeled coherent vs. incoherent, depending on whether these actions were based on prior actions [15]. Finally, the same consecutive actions were collapsed into a single action but labeled multiple. For example, two consecutive short reads were collapsed into an action named multiple short read.

3.3 Applying ESB and TIPS

Traditional sequence mining was applied to the preprocessed dataset to get frequent sequential patterns. The threshold for the support value was 0.5. The maximum gap was 2. This step resulted in 176 frequent sequential patterns. Then, each student's preprocessed log file was segmented into five bins of equal size. For each frequent sequential pattern, its occurrences were calculated per bin per student. Next, Kendall's W and IG of each pattern were computed. These patterns were ranked based on Kendall's W and IG, respectively.

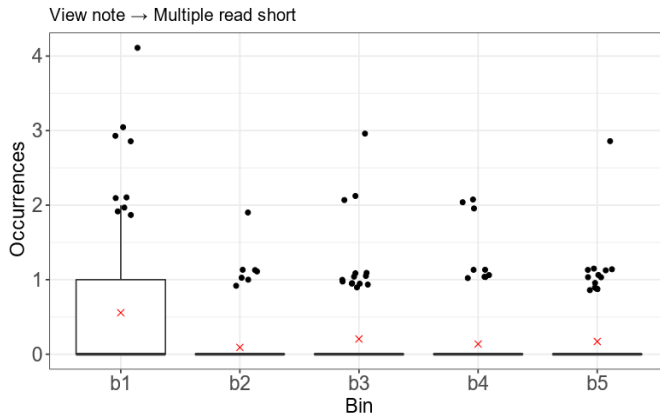
3.4 Results

Some patterns had a high W-based ranking but a comparably lower IG-based ranking or a high IG-based ranking but a comparably lower W-based ranking. Table 1 presents the ranking, Kendall's W, and IG of four such patterns. *View notes* → *Multiple short read* and *Read short* → *Multiple incoherent read* were ranked in the top 10

based on Kendall's W, but 33rd and 35th based on IG. In contrast, *Short read* → *Coherent read* and *Taking a quiz* → *Prompt* → *Coherent revision* were ranked in the top 10 based on IG, but 37th and 39th based on Kendall's W.

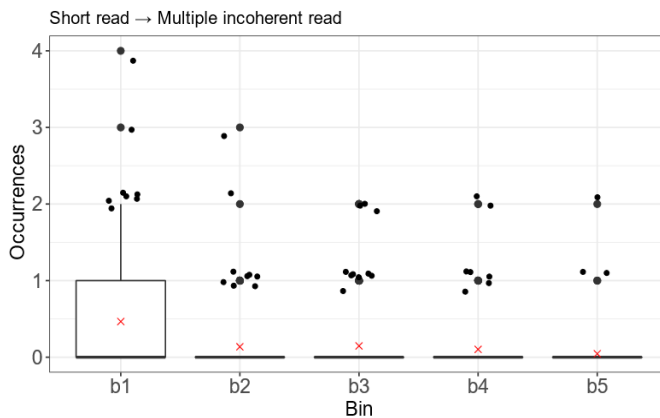
Figures 1 to 4 use boxplots to display the occurrences of the four patterns in each bin. The evolutions of *View note* → *Multiple short read* and *Short read* → *Multiple incoherent read* was quite similar. Both their usage was more frequent in the first bin than in the others and had little variation among the four last bins. Forty percent of students made *View note* → *Multiple short read* in the first bin, while less than 16% of students made this pattern in the other bins. Similarly, thirty-four percent of students executed *Short read* → *Multiple incoherent read* in the first bin, but less than 12.5% of students did so in the others.

By contrast, *Short read* → *Coherent read* and *Taking a quiz* → *Prompts* → *Coherent revision* were less frequent in the first bin than the others. Thirty percent of students executed *Short read* → *Coherent read* in the first bin, but over 55% of students did so in the others. Twenty-five percent of students made *Taking a quiz* → *Prompts* → *Coherent revision* in the first bin, but over 40% of students did so in the others.



Note. 'x' indicates the mean. Dots are outliers within a bin, i.e., cases whose occurrences greater than 'the median + 1.5 * IQR' (distance between the first and third quartiles). Dots are jittered.

Figure 1. The boxplot of the occurrences of *View note* → *Multiple short read*.



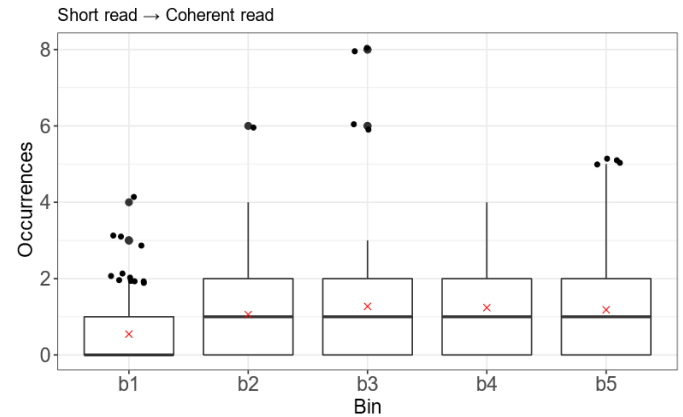
Note. 'x' indicates the mean. Dots are outliers within a bin, i.e., cases whose occurrences greater than 'the median + 1.5 * IQR' (distance between the first and third quartiles). Dots are jittered.

Figure 2. The boxplot of the occurrences of *Short read* → *Multiple incoherent read*.

There are also similarities between ESB and TIPS. For instance, fourteen patterns were ranked in the top 20 most interesting patterns by both Kendall's W and IG, and ten patterns were ranked in the lowest 20 by both metrics.

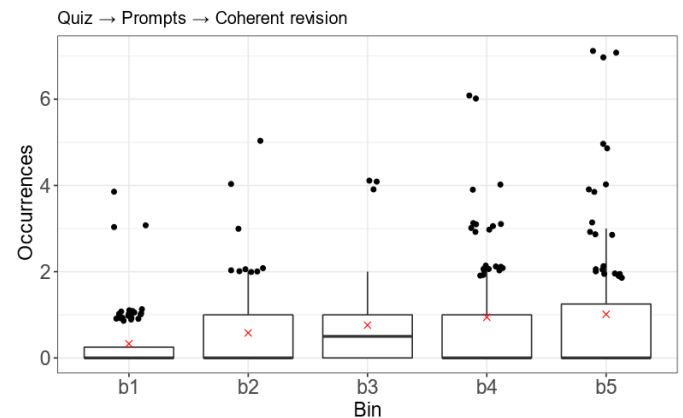
Table 1. Four selected sequential patterns.

Pattern	W - ranking	IG - ranking	Kendall's W	IG
View note → Multiple short read	10	33	0.117	0.051
Read short → Multiple incoherent read	8	35	0.133	0.049
Short read → Coherent read	37	10	0.062	0.073
Taking a quiz → Prompts → Coherent revision	39	7	0.061	0.078



Note. 'x' indicates the mean. Dots are outliers within a bin, i.e., cases whose occurrences greater than 'the median + 1.5 * IQR' (distance between the first and third quartiles). Dots are jittered.

Figure 3. The boxplot of the occurrences of *Short read* → *Coherent read*.



Note. 'x' indicates the mean. Dots are outliers within a bin, i.e., cases whose occurrences greater than 'the median + 1.5 * IQR' (distance between the first and third quartiles). Dots are jittered.

Figure 4. The boxplot of the occurrences of *Taking a quiz* → *Prompts* → *Coherent revision*.

4. Discussion

This paper highlighted three differences between ESB and TIPS. The first one is that the implicit assumption of ESB may be more natural than TIPS. ESB assumes that the bin number (i.e., time) influences the occurrences of a pattern, while TIPS implies that the occurrences of a pattern influence the bin number (see section 2.2).

The second difference is the interpretability. It is easier to interpret the ESB metric (i.e., effect size) than the TIPS metric (i.e., IG). For instance, the Kendall's W of *View note* → *Multiple short read* and *Short read* → *Multiple incoherent read* was 0.117 and 0.133, respectively, and their IG were 0.051 and 0.049, respectively. A Kendall's W greater than 0.1 but smaller than 0.3 means a small effect [3], so the two patterns have small variation across time. However, it is hard to understand what an IG of 0.051 or 0.049 means as both the number of bins and the number of distinct values of pattern occurrences may influence the range of IG.

The results of the application example revealed the third difference: the rankings of sequential pattern based on the effect size and IG were different. This difference is understandable because the formulas for the effect size and information gain are quite different.

Based on Kendall's W, sequential patterns with more occurrences in the first bin and few occurrences in the others were ranked higher than patterns with fewer occurrences in the first bin and more occurrences in the others. By contrast, based on IG, the former was ranked lower than the latter.

Although for all the above sequential patterns, there is a big difference in pattern usage between the first bin and the others, Kendall's W prefers *View note* → *Multiple short read* and *Short read* → *Multiple incoherent read* because the variation of their occurrences across students (between-student variation) were small within each of bin 2 to 5. Recall that less than 16% and 12.5% of students made these patterns in bin 2 to 5, respectively. This means that most of their occurrence values were zero in bin 2 to 5. In contrast, many occurrence values of *Short read* → *Coherent read* and *Taking a quiz* → *Prompts* → *Coherent revision* in bin 2 to 5 was non-zero (over 55% and 40% of students did them, respectively), and their usage had higher variation within each of bin 2 to 5 than *View note* → *Multiple short read* and *Short read* → *Multiple incoherent read* (see Figure 1 to 4). In one-way repeated ANOVA, the between-student variation is an error term. Considering the error term, the variation across bins were higher for *View note* → *Multiple short read* and *Short read* → *Multiple incoherent read* than for *Short read* → *Coherent read* and *Taking a quiz* → *Prompts* → *Coherent revision*. Therefore, the former patterns had a greater Kendall's W than the latter.

IG prefers *Short read* → *Coherent read* and *Taking a quiz* → *Prompts* → *Coherent revision* because many of their occurrence values in bin 2 to 5 were non-zero, and the distribution of these non-zero values varied across bins. For example, the number of students that did *Short read* → *Coherent read* two times was the biggest bin 2, but the number of students that did this pattern four times was the biggest in bin 4. Knowing this occurrence differences of *Short read* → *Coherent read* among bin 2 to 5 could decrease the uncertainty about the bin number (the label). However, most occurrence values of *View note* → *Multiple short read* and *Short read* → *Multiple incoherent read* were zero in bin 2 to 5. Knowing their occurrences provided less information about the bin number than knowing the occurrences of *Short read* → *Coherent read* and *Taking a quiz* → *Prompts* → *Coherent revision*.

In summary, the ESB approach may assign higher rankings than TIPS to patterns with more occurrences in one bin but few and similar occurrences in the others, while the latter may assign higher rankings than the former to patterns with fewer occurrences in one bin but more and similar occurrences in the else.

Thus, ESB would be useful if the goal is to identify sequential patterns that mainly appear in only one bin. Such patterns may inform the intervention and learning design. For instance, both *View note* → *Multiple short read* and *Short read* → *Multiple incoherent read*, patterns that mainly occurred in the first bin, are generally considered as bad strategies in Betty's Brain. This suggests that students might not be familiar with how to utilize the resource page when they start using Betty's Brain to learn climate change. Therefore, the training session may need to teach students more about how to read the resource page effectively.

4.1 Next steps

The application of TIPS and ESB to the example data provided initial insights about the relative advantages of these approaches, but it is necessary to obtain a more comprehensive understanding of their differences in ranking sequential patterns. This goal will be achieved by conducting a larger scale investigation where TIPS and ESB will be applied to dataset from various learning environments. Such investigation will demonstrate under which situation one method has better utilities than the other so that researchers can make an informed decision about which approach is most appropriate given a research purpose.

While our preliminary application example suggests the utility of ESB to provide insights into improving learning intervention, the goal of the current paper was to propose a new methodological approach for mining temporally interesting sequential patterns. As such, further work will be necessary to leverage ESB to answer formal research questions, such as whether an intervention is effective [7].

5. REFERENCES

- [1] Baker, R. 2010. Data mining for education. in McGaw, B., In *International Encyclopedia of Education*, Peterson, P. and Baker, E, Ed. Elsevier Ltd, Oxford, UK, 112-118.
- [2] Biswas, G., Segedy, J. R., & Bunchongchit, K. 2016. From design to implementation to practice a learning by teaching system: Betty's brain. *International Journal of Artificial Intelligence in Education*, 26(1), 350-364.
- [3] Field, A. 2013. *Discovering statistics using IBM SPSS statistics*. SAGE, 2013.
- [4] Kang, J., Liu, M. and Qu, W. 2017. Using gameplay data to examine learning behavior patterns in a serious game. *Computers in Human Behavior*, 72. 757-770.
- [5] Kinnebrew, J., Mack, D. and Biswas, G., 2013. Mining temporally-interesting learning behavior patterns. In *Proceedings of the 6th International Conference on Educational Data Mining*, (Memphis, TN USA, July 6-9, 2013). EDM '13.
- [6] Kinnebrew, J.S., Loretz, K.M. and Biswas, G. 2013. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, 5(1). 190-219.
- [7] Kinnebrew, J.S., Segedy, J.R. and Biswas, G. 2014. Analyzing the temporal evolution of students' behaviors in open-ended learning environments. *Metacognition and Learning*, 9(2). 187-215.

- [8] Lakens, D. 2013. Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and ANOVAs. *Frontiers in Psychology*, 4, 863.
- [9] Merceron, A. and Yacef, K., 2008. Interestingness measures for association rules in educational data. In *Proceedings of the 1st International Conference on Educational Data Mining*, (Montreal, QC Canada, June 20-21, 2008). EDM '08.
- [10] Moon, J. and Liu, Z. 2019. Rich Representations for Analyzing Learning Trajectories: Systematic Review on Sequential Data Analytics in Game-Based Learning Research. In *Data Analytics Approaches in Educational Games and Gamification Systems*, Tlili A., Chang M, Ed. Springer, Singapore, 27-53.
- [11] Molenaar, I. (2014). Advances in Temporal Analysis in Learning and Instruction. *Frontline Learning Research*, 2(4), 15-24
- [12] Molenaar, I., & Järvelä, S. (2014). Sequential and temporal characteristics of self and socially regulated learning. *Metacognition and Learning*, 9(2), 75-85
- [13] Myers, J.L., Well, A. and Lorch, R.F. 2010. *Research design and statistical analysis*. Routledge, 2010.
- [14] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81-106.
- [15] Segedy, J.R., Kinnebrew, J.S. and Biswas, G. 2015. Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments. *Journal of Learning Analytics*, 2(1). 13-48.
- [16] Taub, M. and Azevedo, R. 2018. Using Sequence Mining to Analyze Metacognitive Monitoring and Scientific Inquiry based on Levels of Efficiency and Emotions during Game-Based Learning. *Journal of Educational Data Mining*, 10(3). 1-26.
- [17] Tomczak, M. and Tomczak, E. 2014. The need to report effect size estimates revisited. An overview of some recommended measures of effect size. *TRENDS in Sport Sciences*, 21(1). 19-25.
- [18] Van Laer, S. and Elen, J. 2018. Towards a Methodological Framework for Sequence Analysis in the Field of Self-Regulated Learning. *Frontline Learning Research*, 6(3). 228-249.

Dynamic knowledge tracing through data driven recency weights

Deepak Agarwal
deepakagarwal39@gmail.com

Ryan S. Baker
University of Pennsylvania,
Philadelphia PA, USA
rybaker@upenn.edu

Anupama Muraleedharan
Educational Initiatives, India
anupama.muraleedharan@ei-india.com

ABSTRACT

There has been considerable interest in techniques for modelling student learning across practice problems to drive real-time adaptive learning, with particular focus on variants of the classic Bayesian Knowledge Tracing (BKT) model proposed by Corbett & Anderson, 1995. Over time researches have proposed many variants of BKT with differentiation based on their treatment of the underlying parameters: (a) general across student and questions; (b) individualized for students; and (c) individualized for questions. Yet at the same time, most of these variants are similar in that they utilize the same Hidden Markov (HMM) architecture to model student learning and share many of the same drawbacks, including less effective balancing between recent and historical student data and assuming that students learn at the same rate across all the attempts irrespective of if they get the question right. At the same time, these variants share the virtue of parameter interpretability, a virtue not seen in recent efforts to re-cast knowledge tracing as a deep learning problem.

This paper proposes a different architecture that replaces learning rate with recency weights which capture student improvement wholly through data rather than assuming constant learning across attempts and manages recent and historical data more appropriately while retaining the interpretability of BKT parameters. The proposed model was tested on multiple public datasets from ASSISTments and Mindspark and performed similarly to classic BKT model on unseen data.

Keywords

Intelligent tutoring system, Bayesian Knowledge Tracing, Student modelling, Hidden Markov Model (HMM)

1. INTRODUCTION

One of the most common forms of adaptivity in intelligent tutoring systems is mastery learning, where a system provides content on a skill until a student demonstrates they know the skill [8]. Most intelligent tutoring systems rely on “Knowledge Tracing” models which predict whether a student has learned a skill or not based on the interactions with the learning resources related to that skill within the tutoring system. Currently, most systems used at scale rely on Corbett and Anderson’s (1995)

Bayesian Knowledge Tracing (BKT) model or a close variant of it. Most of these models differ in their treatment of the parameters L_0 , G , S and T , but leave the basic structure of the underlying HMM model unchanged, and thus share many of the limitations and drawbacks of the BKT model (e.g. [7, 10, 9, 10]). Recently there have been some attempts to use deep learning-based models in education, termed Deep Knowledge Tracing (DKT) [6, 5]. Though DKT models have performance advantages over BKT, it is extremely difficult to interpret the implicit knowledge model. Khajah and colleagues [6] found that it is possible to make meaningful enhancements to BKT that bring its performance to the same level as DKT models.

In this paper, we propose an algorithm, MS-BKT (Multistate BKT) to address two particular shortcomings of the classic BKT model. First, BKT assumes a constant learning rate after each practice opportunity, irrespective of the student responses, which can lead to bias in estimating student mastery level. Second, BKT represents latent student knowledge as a binary variable with known and unknown states, which is a simplification and assumes that the probability of being in a state at step n depends only on the previous step $n-1$. We suspect that these assumptions limit the BKT model from considering the entire history of responses for students in a balanced manner by giving unproportionately high weight to the most recent attempt. The MS-BKT addresses these issues through two modifications:

- The MS-BKT model gives more weight to recent responses over older ones during the iterative Bayesian update in order to capture changes in student mastery level from data and excludes learning rate T so there is no assumption of fixed learning after each attempt. Please note that this paper uses ‘Recency’ weights differently than previous papers such as Galyardt & Goldin [3] or Gong et al., [4], where they used a decay function to down-weight the older attempts. In comparison, this paper incrementally increases the weight of the newer attempts.
- MS-BKT expands the knowledge node from the typical 2 states (‘Not learned’, ‘Learned’) to 21 states. Adding multiple states to the knowledge node allows MS-BKT to better capture complex sequences of correct and incorrect responses as multiple states make it possible to fine tune the knowledge level more granularly after each new observation than the 2 state model. Given that real world data can be very noisy, MS-BKT model estimates lead to smoother learning curves than classic BKT models.

Deepak Agarwal, Ryan Baker and Anupama Muraleedharan "Dynamic knowledge tracing through data driven recency weights" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 725 - 729

2. APPROACH

2.1 Classic BKT Model Architecture

Classic BKT employs a Hidden Markov Model (HMM) with a two-state ('Not learned', 'Learned') latent node representing student mastery level of the skill and a binary observed node indicating whether the student solved the question correctly or incorrectly as shown in Figure 1. The model assumes that the student can make the transition from not knowing the skill to knowing after every practice opportunity, fit as the learning probability $p(T)$. The model also incorporates the probability that the student may answer a question incorrectly despite knowing the skill (called slip) or may get the answer correct despite not knowing the skill (called guess).

The probability that the student knows the skill gets updated after every practice opportunity through the following equations –

$$p(L_{n-1} | C_n=1) = \frac{p(L_{n-1}) * (1-p(S))}{p(L_{n-1}) * (1-p(S)) + (1-p(L_{n-1})) * p(G)}$$

$$p(L_{n-1} | C_n=0) = \frac{p(L_{n-1}) * p(S)}{p(L_{n-1}) * p(S) + (1-p(L_{n-1})) * (1-p(G))}$$

$$p(C_n | L_{n-1}) = p(L_{n-1}) * (1 - p(S)) + (1 - p(L_{n-1})) * p(G)$$

$$p(L_n) = p(L_{n-1} | C_n) + (1 - p(L_{n-1} | C_n)) * p(T)$$

2.2 Multistate BKT Model Architecture

The architecture for MS-BKT, shown in Figure 2, is similar to that of classic BKT with two changes:

- The "knowledge node" consists of 21 states instead of 2 (Knowledge states are denoted by L_n^i where i is in range

0 to 20 and $\sum_i p(L_n^i) = 1$). 21 discrete states were selected as it was granular enough to give a precise estimate with manageable calculation overhead. The choice of number of states can be explored further in future work, including the possibility of a continuous distribution function.

- A recency weight parameter R is introduced in place of the transition probability $p(T)$. The model assigns a default weight of 1 to the first attempt and thereafter weight increases incrementally by a fixed quantum R for each new attempt. The optimal value of R can be learnt from data. Recent attempts are incrementally weighted more based on the intuition that the recent data will reflect current learning level better but at the same time, older attempts cannot be ignored completely as data can be inherently noisy.

This effectively means that MS-BKT is the same as classic BKT in that new data is integrated with a past estimate aggregating all past data, but differs in that the past estimate is now a distribution and that the weight of the new data increases over time.

$p(L_n)$ = Probability that the skill is known at n^{th} attempt
 $p(T)$ = Probability that the skill will be learnt at next opportunity
 $p(G)$ = Probability of guess
 $p(S)$ = Probability of slip

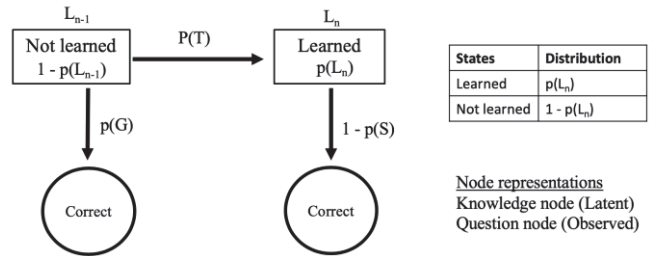


Figure 1. Classic BKT model

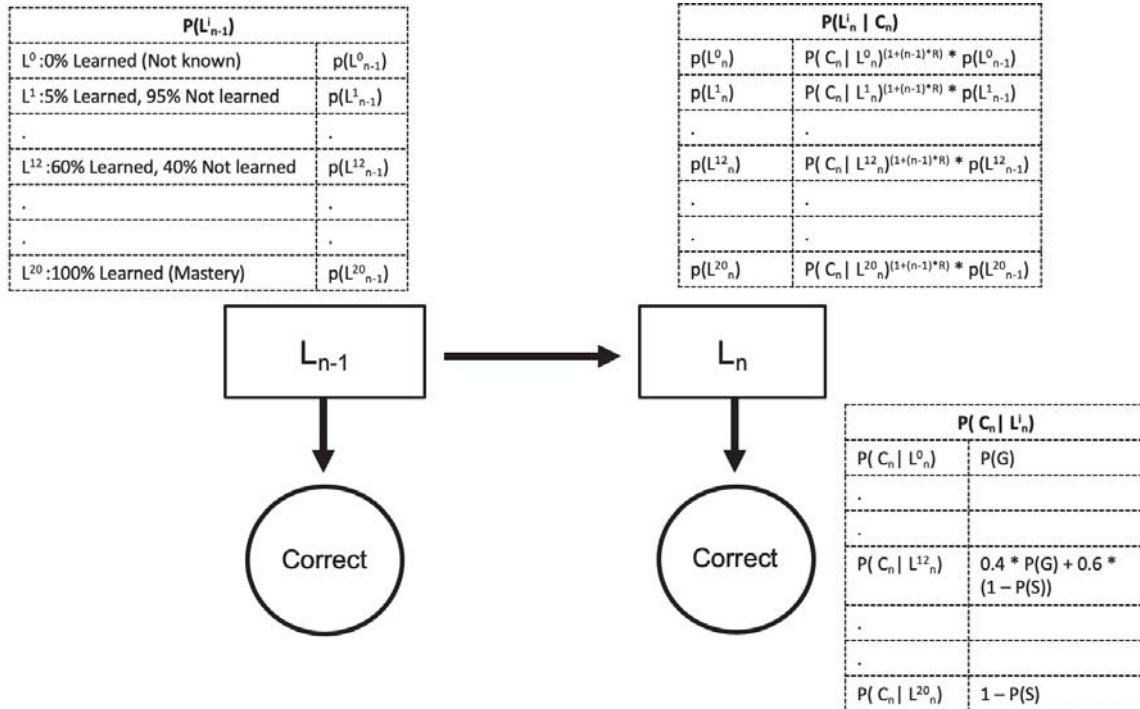


Figure 2. MS-BKT model architecture.

2.3 Updating Student Knowledge

Given an observation of the student's response at time opportunity n (correct or incorrect), updated student knowledge (L_n) is calculated using Bayes' rule. Since L_n now consist of 21 states, the probability of each state needs to be updated after every new observation as follows:

$$p(L_n^i | C_n) = \frac{p(C_n | L_{n-1}^i)^{(1+(n-1)*R)} * p(L_{n-1}^i)}{\alpha(\text{Normalizing factor})} \quad \text{For } i \text{ in range } 0 \text{ to } 20$$

Where:

- $p(L_n^i | C_n)$ represents the probability of the i^{th} knowledge state given the observation C_n
- $p(C_n | L_{n-1}^i)$ is the likelihood factor. $p(C_n | L_{n-1}^i) = L_{n-1}^i * (1 - p(S)) + (1 - L_{n-1}^i) * p(G)$
- $p(L_{n-1}^i)$ is the prior probability of the i^{th} knowledge state
- $1 + (n-1)*R$ is the weight for the n^{th} response, where n is the number of actions so far and R is a free parameter estimated during model fitting
- α is the normalizing factor which is computed at each iteration to be the value that ensures that probabilities across all the 21 states sum to 1

Once new probabilities are calculated, L_n value is estimated using maximum a posteriori probability (MAP) estimate that equals the mode of the posterior distribution. The advantage of using a MAP estimate over an EAP estimate is that it provides sharper updates

even at the initial responses stage. The overall model parameters are learned from data using 'Expectation Maximization'.

3. RECENCY WEIGHTS SUCCESSFULLY CAPTURES REAL TIME LEARNING FROM DATA

In this section we use a hypothetical example to show that the MS-BKT model is capable of capturing learning and forgetting from data itself by the property of recency weights and does not need an external fixed amount of learning to be added after each attempt, unlike classic BKT. This example tracks how the mastery level of three fictitious students changes as they attempt 10 questions on a skill for MS-BKT model. Parameter values used for the below illustration are as follows: L_0 : 0.5; G : 0.1; S : 0.1; and T : 0.3.

All three students answer five questions out of 10 correctly, but their patterns are different. Student1 answers questions correctly and incorrectly consecutively. Student2 answers more questions correctly in later attempts, whereas for Student3 the situation is reversed, suggesting that Student2 displays a learning behavior whereas Student3 displays forgetting.

As the following table shows, the mastery level estimate from MS-BKT for Student2 (pattern with learning) is considerably higher than for Student3 (pattern with forgetting), though both students answer 5 out of 10 questions correctly. The mastery estimate of Student1, which was added as a base case, is close to 0.5 as expected.

Table 1. Response patterns used for generating posterior distribution curves

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Mastery Estimate
Student1	0	1	0	1	0	1	0	1	0	1	0.55
Student2	0	0	0	0	1	0	1	1	1	1	0.67
Student3	1	1	1	0	1	0	0	1	0	0	0.37

4. OTHER OBSERVATIONS

In the BKT model, L_n values get updated very aggressively after each observation and result in large fluctuations in the value of L_n (though, with reasonable parameter values, BKT still has lower fluctuation than has been reported for DKT, e.g. Yeung & Yeung, 2018). In comparison to classic BKT model, the MS-BKT model does not fluctuate that widely for the same set of skill parameters. MS-BKT model also takes in account the entire history of the student's responses in a more balanced manner whereas in BKT, a student's response history prior to the third or fourth attempt may become irrelevant due to aggressive updates.

Table 2 and Figure 3 illustrate the above two points using fictitious student data. The underlying BKT and MS-BKT models use the same parameter values for L_0 , G , and S ; L_0 : 0.5, G : 0.1, and S : 0.1. T value for BKT model is 0.1 and R value for MS-BKT is 0.3. The comparison of L_n values for Student4, Student5, and Student6 show that L_n values have significantly higher fluctuations for BKT model in comparison to MS-BKT model. Also, in the cases of Student4 and Student7, L_n estimates are

extremely high for the BKT model and does not correspond to the respective response patterns. For Student4, L_n shoots up drastically to 0.75, even though there is a long history of incorrect responses on previous attempts and learning rate is only 0.1. By comparison, the L_n value is around 0.30 for the MS-BKT model. For Student7, L_n value is 0.83 in the case of the BKT model even though 3 out of last 4 responses were incorrect. This is largely due to the fact that the BKT model considers fixed learning rate irrespective of the student responses. The same L_n value for the MS-BKT model is 0.45, as the model is able to derive learning or forgetting directly from the data. Comparison of the response patterns of Student5 and Student6 shows some trade-offs between models. MS-BKT model estimates the L_n value to be 0.55 for Student6 in comparison to 0.90 estimated by the classic BKT model – probably a better fit, since the student has alternated between answering the questions correctly and incorrectly. By contrast, for student5 MS-BKT estimates L_n value to be 0.70 giving the student more credit as for recent responses being correct – perhaps a little too low compared to BKT. Of course, all of these estimates can be adjusted by tuning the parameters during model development.

Table 2. Response patterns used for comparing the two models

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Average	BKT	MS-BKT
Student4	0	0	0	0	0	1	0	1	0.25	0.75	0.30
Student5	0	0	0	0	1	1	1	1	0.50	1.00	0.70
Student6	0	1	0	1	0	1	0	1	0.50	0.90	0.55
Student7	0	1	1	1	0	1	0	0	0.50	0.83	0.45

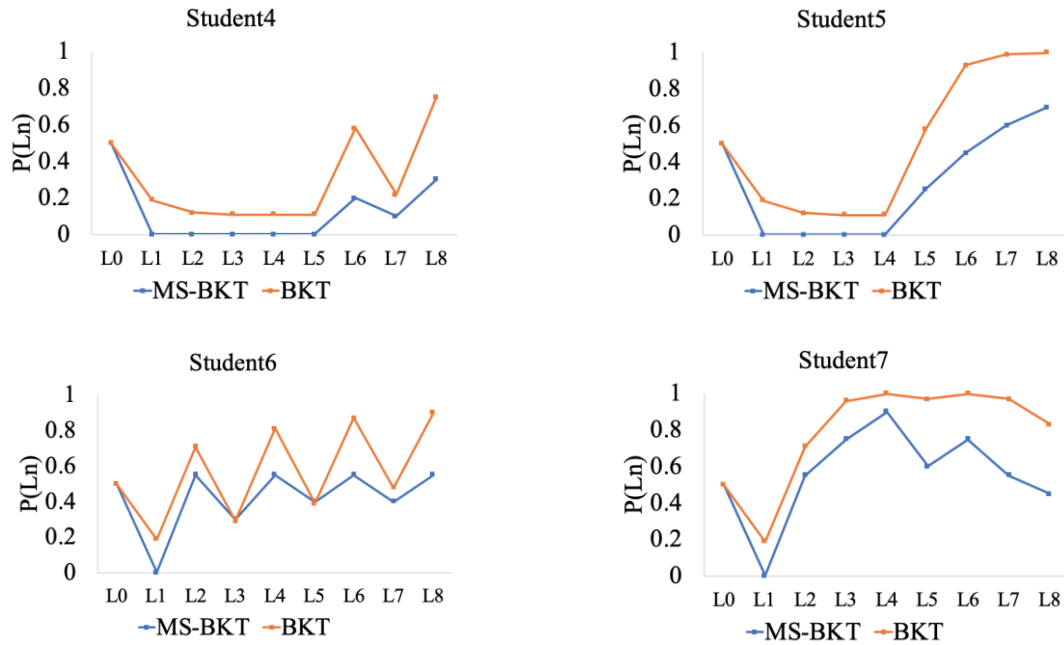


Figure 3. Comparison of L_n estimate for BKT and MS-BKT.

Table 3. L_0 , G, S, T values for BKT and MS-BKT models

Dataset	#Student	#Attempts	BKT				MS-BKT			
			L_0	G	S	T	L_0	G	S	T
G6_207	620	6	0.42	0.28	0.15	0.08	0.56	0.27	0.29	0.25
G7_233	540	7	0.73	0.26	0.22	0.01	0.65	0.09	0.25	0.25
G6_217	500	5	0.61	0.30	0.13	0.10	0.60	0.29	0.21	0.25
PER015	855	5	0.50	0.11	0.30	0.15	0.58	0.15	0.29	0.25
WNO021_57	536	6	0.80	0.24	0.18	0.11	0.66	0.27	0.19	0.50
WNO021_48	536	6	0.78	0.30	0.08	0.30	0.74	0.29	0.09	0.25

Table 4. Comparison of BKT and MS-BKT models

Dataset	#Students	#Attempts	BKT		MS-BKT	
			AUC ROC	RMSE	AUC ROC	RMSE
G6_207	156	6	0.707	0.457	0.712	0.460
G7_233	138	7	0.663	0.464	0.640	0.468
G6_217	126	5	0.664	0.442	0.650	0.446
PER015	171	5	0.659	0.480	0.652	0.483
WNO021_57	134	6	0.618	0.421	0.639	0.425
WNO021_48	134	6	0.702	0.337	0.664	0.345

5. PREDICTION QUALITY

We used 6 datasets across 2 different ITS (Assistments - G6_207, G7_233, G6_217; Mindspark - PER015, WNO021_57, WNO021_48) to compare the performance of the MS-BKT model against classic BKT model. Mindspark is an adaptive online tutor for Math and English, developed by Educational Initiatives (EI). Mindspark Math currently has 80,000 users across India, primarily from private schools, in grades 1 to 9. ASSISTments is an online tutor that supports student learning through the use of scaffolding, hints, and immediate feedback. All the datasets consist of student responses in the form of correct or incorrect answers from specific problems tagged by skill. The performance was compared on a hold-out data set consisting of 20% of the data. Table 3 lists out the parameter values for the two models for all the datasets using training data. The parameters for each model were tuned using the simple Brute Force approach. Table 4 compares the performance of both the models on hold-out dataset. Results show that the classic BKT model performs better than MS-BKT model on most of the datasets (except G6_207 and WNO021_57) but the differences are not very large.

6. CONCLUSION

This paper highlights two issues related to the classic BKT model and tries to address them by proposing a new model (MS-BKT). The paper demonstrates that applying a recency adjustment to Bayesian updates can lead to better properties of knowledge estimation, compared to using a static learning rate. The paper also proposes considering latent student knowledge as a multistate variable instead of 2 states, leading to smoother updates in the learning level estimate. In summary, the MS-BKT model displays some useful properties that are worth considering. Ultimately, models should both capture data well and have desirable properties for actual use, whether for use in a running system or discovery with models analysis. There is considerable future work to be done in refining the MS-BKT model further – such as selection of the appropriate number of knowledge states, implementation of recency weights, and effective ways to tune the model parameters.

7. REFERENCES

- [1] Baker, R.S., Gowda, S.M., & Salamin, E. 2018. Modeling the learning that takes place between online

assessments. *Proceedings of the 26th International Conference on Computers in Education*, 21-28.

- [2] Falakmasir, M. H., Yudelson, M., Ritter, S., & Koedinger, K. 2015. Spectral Bayesian knowledge tracing. *Proceedings of the 8th International Conference on Educational Data Mining*, Madrid, Spain, 360-364.
- [3] Galyardt, A., & Goldin, I. 2015. Move your lamp post: Recent data reflects learner knowledge better than older data. *Journal of Educational Data Mining*. 7, 2, 83-108.
- [4] Gong, Y., Beck, J. E., & Heffernan, N. T. 2011. How to construct more accurate student models: Comparing and optimizing knowledge tracing and performance factor analysis. *International Journal of Artificial Intelligence in Education*. 21, 1, 27-46.
- [5] Jiang, B., Ye, Y., & Zhang, H. 2018. Knowledge tracing within single programming exercise using process data. *Proceedings of the 26th International Conference on Computers in Education*. 89-94.
- [6] Khajah, M., Lindsey, R. V., & Mozer, M. C. 2016. How deep is knowledge tracing? *Proceedings of the 9th International Conference on Educational Data Mining*. 94-101.
- [7] Pardos, Z. A., & Heffernan, N. T. 2011. KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. *User Modeling Adaptation and Personalization Lecture Notes in Computer Science*, 243-254.
- [8] Ritter, S., Yudelson, M., Fancsali, S. E., & Berman, S. R. (2016, April). How mastery learning works at scale. *Proceedings of the Third ACM Conference on Learning @ Scale*. 71-79.
- [9] Wang, Y., & Beck, J. 2013. Class vs. student in a Bayesian network student model. *Artificial Intelligence in Education. AIED 2013*. Lecture Notes in Computer Science, vol. 7926.
- [10] Yudelson, M.V., Koedinger, K.R., & Gordon, G.J. 2013. Individualized Bayesian knowledge tracing models. *International Journal of Artificial Intelligence in Education*. 171-180.

Auto Generation of Diagnostic Assessments and their Quality Evaluation

Soma Dhavala, Chirag Bhatia, Joy Bose, Keyur Faldu, Aditi Avasthi
Embibe Inc.
Bangalore, India
{soma.dhavala, chirag.bhatia, joy.bose, k, a}@embibe.com

ABSTRACT

A good diagnostic assessment is one that can (i) discriminate between students of different abilities for a given skill set, (ii) be consistent with ground truth data and (iii) achieve this with as few assessment questions as possible. In this paper, we explore a method to meet these objectives. This is achieved by selecting questions from a question database and assembling them to create a diagnostic test paper according to a given configurable policy. We consider policies based on multiple attributes of the questions such as discrimination ability and behavioral parameters, as well as a baseline policy. We develop metrics to evaluate the policies and perform the evaluation using historical student attempt data on assessments conducted on an online learning platform, as well as on a pilot test on the platform administered to a subset of users. We are able to estimate student abilities 40% better with a diagnostic test as compared to baseline policy, with questions derived from a larger dataset. Further, empirical data from a pilot gave an 18% higher spread, denoting better discrimination, for our diagnostic test compared to the baseline test.

Keywords

Diagnostic Test Paper, Question Paper Generation, Item Response Theory, Quality Evaluation

1. INTRODUCTION

Learning theory is an important field of research, which incorporates insights from such diverse fields as psychology, pedagogy, neuroscience, and computing to model how well a student learns the taught information. Insights from learning theory are applicable in a wide variety of applications, such as creating intelligent tutor systems and learning platforms, designing courses, designing test papers for exams, and teaching a learner a skill. A prerequisite for any of these activities is to diagnose the current skill level of a new student. This is akin to the *cold start problem* in recommender systems. One proven technique to assessing the current skill

level of a new student is to use a set of assessment challenges, most commonly taking the form of a test paper. A good test paper is one that has specific characteristics in terms of *accuracy* and *discrimination*: The test paper should be able to *accurately* diagnose the ability level of a student for the skill set being evaluated, and it should be able to *discriminate* between students of different abilities. Additionally, it should be able to meet these objectives using as few questions as possible.

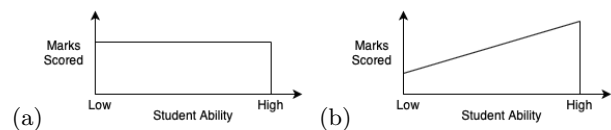


Figure 1: (a) A test in which students of different abilities perform similarly, i.e. get similar scores, is not a good test (b) A better test which can discriminate between students of different ability

If a test paper has questions that many, or all, students answer equally correct or wrong, it will not provide any meaningful information about students. An ideal test paper would reflect student performance such that students with low ability level would get fewer questions correct (lower marks scored) while students with high ability level would get more questions correct (higher marks scored) Fig. 1 illustrates both types of test papers.

In this paper, we present an approach to select questions from a question bank, using configurable policies, that meet the above criteria. We use the selected questions to create a test paper. We then evaluate the generated test paper as per the criteria of accuracy and discrimination, and thus decide on the *goodness* of the policy. Finally, we validate the generated test paper with the best policy on a pilot study of students attempting the test paper. The rest of the paper is organized as follows. Section 2 looks at related work in test paper generation. Section 3 describes our approach to model the problem. Section 4 outlines multiple policies to select questions to compose a test paper. Section 5 discusses the quality evaluation criteria. Section 6 discusses and analyses the results on the simulated and pilot test papers. Finally, Section 7 concludes the paper and presents directions for future work.

2. RELATED WORK

Soma Dhavala, Chirag Bhatia, Joy Bose, Keyur Faldu and Aditi Avasthi "Auto generation of diagnostic assessments and their quality evaluation" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 730 - 735

Cen et al. [1] described the architecture of an automated test generation system, using random selection and other strategies to generate questions. They focused on the architecture and not on the effectiveness of the selected questions in diagnosing student ability. A number of studies have been performed on the effectiveness of adaptive test generation, using algorithms to select test questions dynamically from a given pool. Linacre [2] surveyed computer adaptive testing (CAT) in relation to its history and advantages such as needing fewer questions and a shorter time frame than classical tests to diagnose a student's skill level. The questions are selected from a question database, and models such as the Rasch model (a variant of the popular item response theory (IRT) model [3]) are used. CAT starts by presenting questions with average calibrated difficulty at first, then increasing or decreasing the difficulty level of subsequent questions depending on whether the student got the answer right or not. This continues until the system has reached a good estimate of the student's true ability. CAT testing has limitations such as restrictions on re-calibration if the student changes their mind about a previous answer. Another limitation is that the calibration methodology is based on a single parameter, that of difficulty, and not other parameters such as behavior. Kingsbury [4] suggested an approach to improve the adaptive calibration process in a CAT test by considering the student's momentary trait level estimate, in addition to item difficulty, while selecting questions. Also, the estimated difficulty of each question, initially tagged by experts, is continually calibrated based on how many students have answered correctly in the tests given. They found this approach yielded better results in estimating the difficulty of an item. Makransky [5] compared calibration strategies for test questions, including a random strategy and a strategy where the questions are calibrated at the end of a phase or multiple phases, in order to estimate the item difficulty accurately. They implemented the strategies on 1PL and 2PL models of IRT, and found that a continuous updating strategy performed best. Wim [6] surveyed student ability estimation as well as item selection for CATs, using models such as Maximum Likelihood and Bayesian criteria to estimate ability and mean absolute error as the evaluation parameter. Our paper also uses similar models, and additionally realtime data of administered tests to evaluate the accuracy of the models as well as the discrimination ability.

Some researchers have studied factors other than item difficulty when selecting questions. Liu et al [7] found that behavioral factors such as test-taking motivation in students can play an important role in determining learning outcomes. Similarly, Tsousis [8] suggested a variant of the IRT model in which behavioral parameters like item response time can be incorporated. In another study on behavior as a factor, Jaworski [9] discussed the calibration of control questions in a personalized polygraph test, using emotion and behavior as parameters in selecting the questions. Daroudi et al. [10] surveyed reinforcement learning as a strategy to model the sequencing of instructions in order to maximize learning.

3. PROBLEM FORMULATION

For our analysis, we use a question database taken from Embibe, an online learning platform, along with responses from a set of students on each question. The student's abil-

ity is a latent variable, which when estimated with statistically adequate data samples gives a better estimation of the ground truth. For this paper, we consider the ability derived from a larger dataset (in this case, the question database) as ground truth, and abilities derived from a single test as the predicted abilities. For each question in the database, we have the following parameters: Discrimination factor, Difficulty level, Chapter number (represents the chapter number in the syllabus which the question comes from) and Student behavior data for the question. For each student, we have the Ability and Discrimination factor parameters (from the fitted IRT model). The difficulty level and chapter number of each question are annotated by human experts. The anonymized data related to the student responses is collected by the platform.

Out of this ground truth dataset, our objective is to select a subset of questions to assemble into a test paper, which meets the criteria such as best discriminative ability and best match of the identified student ability with the ground truth.

		Students										
		0	1	2	3	M	α	β	Ch		
Question	0	0	0	1	1	1	0	1	0.5	0.6	2
	1	1	1	0	1	0	1	0	0.7	0.9	3
	2								0.1	0.5	1
	3						

	N						
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							
							

θ : Student Ability
 α : Discrimination
 β : Difficulty
Ch: Chapter

Figure 2: Ground truth dataset of questions taken of a learning platform, with IRT parameters and chapter information

Fig. 2 illustrates the ground truth dataset of questions, along with data on the correctness of students' past responses on each question (whether they answered the question correctly or not). Out of this matrix, we select a small subset of exam questions that can discriminate between students of different abilities.

As per the Item Response Theory (IRT) model, for each question we have a measure of its difficulty and discriminative ability, as well as a measure of the student ability for each student. The standard IRT model gives a relation between the ability and the difficulty, based on one or more parameters and predicts the likelihood that the student will answer that question correctly. We use the 2PL IRT model to calibrate and evaluate our generated test papers.

As per the 2PL IRT model, the probability or likelihood of the student answering a question correctly is given by the following equation:

$$P(X = 1|\theta, \alpha, \beta) = \frac{e^{\alpha(\theta - \beta)}}{1 + e^{\alpha(\theta - \beta)}} \quad (1)$$

Here, θ represents the student's skill/ability level, α represents the discrimination factor of the question, β represents the difficulty level of the question and P represents the probability that the student will answer correctly.

We infer the IRT model parameters (α, β, θ) from our ground truth dataset by fitting a fully connected deep neural network (modeled using Keras [11] library). The inputs to the neural network are one-hot encodings of the student and question vectors, and the output is the correctness of the student's response for that question, which is a binary value. The IRT parameters are estimated by fitting the neural network using Binary Cross Entropy (BCE) loss. The fitted model is scalable and can handle missing data and imbalanced classes very well. Fig. 3 shows the architecture of the deep neural network for 1PL IRT model. Other IRT models can be realized using the same template.

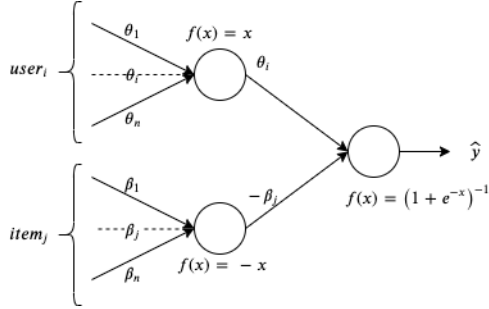


Figure 3: Neural network architecture for estimating 1PL IRT parameter values.

Our problem of selecting an optimal set of questions to form a test, following various constraints, can be modeled in the following manner: Let A be a K -dimensional tensor of size n_1, n_2, \dots, n_K . Each entry of this tensor is either 0 or 1 indicating whether a question with a particular set of attributes was sampled or not. Each dimension of this tensor represents a question attribute, such as a *chapter number*, *difficulty level*, etc. For example, let us say we are interested in creating a test such that four chapters are to be covered, with difficulty levels ranging from 1-10. Then we have $n_1 = 4, n_2 = 10$. Here $A[i, j] = 1$ means that we select a question from chapter i with difficulty level j . We can then set constraints on this tensor to reflect some desired characteristics. For example, the following constraint says that there has to be at least one question from each chapter.

$$\sum_j A[i, j] \geq 1$$

Likewise, we can say that difficulties should follow a certain distribution. Let d_j be the number of questions we like to have whose difficulty level is j . Then,

$$\sum_i A[i, j] = d_j$$

Now we can count how many times the above condition is not met, as a way to measure the quality of the assignment/sampling. Using this, we can form an objective function that evaluates how well the chosen test reflects the above loss, which simply counts the number of disagreements.

$$\min \sum_j I(\sum_i A[i, j] \neq d_j)$$

The above objective function is zero when conditions are met exactly (hard constraint). We can generalize this idea

to include constraints about all the question attributes (that are factor variables). Let there be n_k levels for the k -th dimension of the tensor A . These levels represent, for each attribute, the range of values that attribute can take. Let $d_{k(i)}$ be the number of questions needed where the question's k -th attribute has level $c_{k(i)}$. Notice that different attributes can have different number of levels.

$$\min \sum_{k=1}^K \lambda_k \sum_{i=1}^{n_k} I(\sum A^k[i] \neq d_{k(i)})$$

Here $\sum A^k[i]$ means that, we select the k -th dimension of the tensor, and its i -th cube, and summing along the cube. In particular, when $\forall_{k(i)} d_{k(i)} = 1$ then Latin HyperCube sampling can be used. The above objective can also be used as a fitness function in genetic algorithms or other search techniques, both stochastic and deterministic, to allocate questions to a test paper. λ_k is a weight parameter which we can tune, for our purposes in this paper we set all the values of λ_k to be equal.

The above objective function, which can be coupled with other IRT based test design objectives, is dealing with domain constraints. Test designs that consider the variance-covariance matrices of parameters in the IRT are also widely used[12]. In particular, the relationship between the item difficulty, discrimination and ability has been addressed from a D-optimality sense. Based on those insights, we formulate a theorem along with proof as below. This is used to develop one of our question selection policies.

THEOREM 1. *In a 2PL IRT model, when the difficulty of an item is close to the ability of the person, an item with high discrimination will have high information, and is locally D-optimal.*

PROOF. The Item Information function for the 2PL IRT model introduced earlier is given as:

$$I(\theta; \alpha, \beta) = \frac{\alpha^2 e^{\alpha(\theta-\beta)}}{(1 + e^{\alpha(\theta-\beta)})^2}$$

The above equation can be rewritten as:

$$I(\epsilon; \alpha) = \frac{\alpha^2 e^{\epsilon\alpha}}{(1 + e^{\epsilon\alpha})^2}$$

where $\epsilon = \theta - \beta$. Let us consider another item with higher discrimination $\alpha' = \alpha + \delta, \delta > 0$, but with difficulty close to the ability. Then,

$$\lim_{\epsilon \rightarrow 0} \frac{I(\epsilon; \alpha')}{I(\epsilon; \alpha)} = \left(\frac{\alpha + \delta}{\alpha} \right)^2 > 1$$

Hence, an item with high discrimination will have higher asymptotic relative efficiency, when the difficulty is in the neighbourhood of the ability. We can claim that such a policy is D-optimal. \square

4. TEST PAPER GENERATION

In order to generate a test paper, we propose a set of candidate policies to select questions from the ground truth dataset and assemble the selected questions to form a test paper. All policies assume that the syllabus is covered adequately, i.e. questions are selected from each area of the

syllabus. Based on theorem 1 and [12], we select questions with a mix of difficulty levels. We evaluate these policies as per their effectiveness in distinguishing students. To evaluate each policy, we measure parameters such as spread of the scores obtained by different students on the test and how well the diagnosed abilities of the students correspond with the ground truth abilities (by computing the Mean Square Error, Spearman's Rank Correlation and a scatter plot of diagnosed ability vs. true ability). We then choose the best policy to generate a test paper to validate our model by testing on a real world group of students using the same online learning platform where we sourced the ground truth dataset. Fig. 4 shows a flowchart illustrating this method.

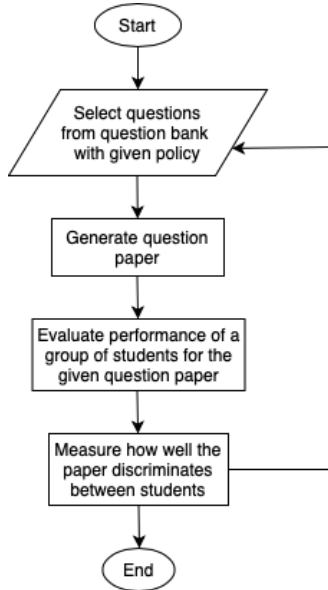


Figure 4: Flowchart showing a method to generate test papers from a question bank by selecting questions using a configurable policy, and evaluating how well the policy diagnoses different kinds of students

The candidate policies are described in the following subsections.

4.1 Baseline policy π_{BSP}

As a baseline, we select N questions from the ground truth dataset, by randomly selecting over other question attributes after ensuring a mix of difficulty levels and syllabus coverage. This selection of questions becomes our standard baseline - **BSP**.

4.2 Discrimination only policy π_{DOP}

We use the discrimination parameter values inferred from the fitted 2PL IRT model. We select questions with a mixture of difficulty levels, and the highest values of the discrimination factor for each given difficulty level. We select N questions from the ground truth dataset, ensuring syllabus coverage (at least one question from each chapter), but also ensuring that the overall discrimination factor of the questions is maximized. This policy, **DOP**, ensures that high discrimination questions are selected, at any given difficulty level.

4.3 Discrimination+behavior policy π_{DBP}

In this policy, we incorporate behavior parameters along with discrimination, difficulty and syllabus coverage, while selecting questions. Behavior parameters refer to the student behavior when taking the test, captured by the learning platform. These include parameters such as number of questions that are likely to be answered too fast and incorrectly, or questions that are answered too slow but correctly, among others. The questions are tagged as per which parameters are mostly manifested by students answering that question and the top questions from each parameter are selected. This policy, **DBP**, ensures that high discrimination questions as well as student behavior are taken into account.

5. QUALITY EVALUATION CRITERIA

In order to evaluate the generated test papers, we use two criteria: *accuracy* and *discrimination*. *Accuracy* refers to how closely the diagnosed ability using the student responses to the test paper corresponds to the actual ability of the students. We use the RMSE between the ground truth and the inferred ability as a measure of the accuracy. The rank correlation between the ground truth rank and the estimated rank, and scatter plot between the inferred and ground truth ability, also indicate the accuracy.

Discrimination measures how successful the test paper is in discriminating between students of different abilities. We evaluate the accuracy and discrimination for the generated test papers on a subset of M students (evaluation student set) from our ground truth dataset. We use the spread and distribution of scores as a measure of the discrimination.

Evaluation using RMSE

Using the IRT model, we predict the probability of each student in the evaluation set answering the questions correctly, and compute the average ability from the scores of the students if they were to attempt the generated test paper. We also determine the ground truth ability of each student from the IRT model. Finally, we compute the root mean squared error (RMSE) between the ground truth ability and inferred ability to get a measure of the accuracy.

Evaluation using Spearman's ρ

Here we sort the abilities of students obtained from the ground truth data and from the generated test, and determine the rank correlation ρ between the two ranks.

Evaluation using scatterplots

We plot the abilities of students, inferred from the ground truth, against the diagnosed abilities from the generated test papers. The degree of scatter gives an indication of how much the ability matches the inferred ability.

6. RESULTS AND DISCUSSION

We have an initial ground truth dataset, obtained from the online learning platform, of close to 1300 questions and 1700 students along with the responses for each of the students on each question, along with the derived IRT parameters. From the dataset, we filter those students who have attempted less than 25% of the questions in each paper, so that we have sufficient data to estimate their abilities.

6.1 Simulated tests

We choose 75 questions from the ground truth dataset for each policy, in effect simulating a test of 75 questions. In selecting the questions, we ensure syllabus coverage. Table 1 shows various test statistics.

Table 1: Comparison of test results from simulated tests generated by the three policies

	BSP	DOP	DBP
No. of students	312	312	312
No. of questions	75	75	75
Max. score possible	300	300	300
Max. score achieved	188	251	218
Min. score achieved	-22	-17	-23
Score at 95th percentile	118.4	148	144.5
Score at 5th percentile	3	4	0
Avg. score achieved	60.80	79.25	77.18

Fig. 5 shows the comparison in spread of student scores for the simulated tests on test papers generated using the three policies.

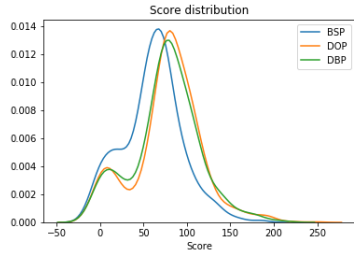


Figure 5: Comparison of the spread in score obtained from the simulated tests generated using following policies: BSP, DOP and DBP.

For each of the test papers selected using different policies, we evaluate the accuracy and discrimination as mentioned in the previous section. We also calculate the ability of each student from the remaining questions in the ground truth dataset, which are not included in any of the generated test papers.

Table 2: Comparison of RMSE (inferred ability and ability from ground truth) and rank correlation ρ in tests generated by different policies

Policies	RMSE	Rank corr ρ
BSP	0.844	0.59
DOP	0.549	0.83
DBP	0.615	0.788

We find that the DOP test gives 24.8% better spread of scores (score at 95th percentile of students - score at 5th percentile), as compared to the BSP baseline. DBP test gives 25.2% better spread. The mean squared error for the inferred ability of the students compared to the ground truth ability is 0.844 for the BSP, 0.549 for the DOP and 0.615 for DBP. Table 2 shows the comparison between the policies

with respect to root mean square error (RMSE) and Spearman rank correlation. We obtain a 40% higher correlation for the DOP policy as compared to BSP.

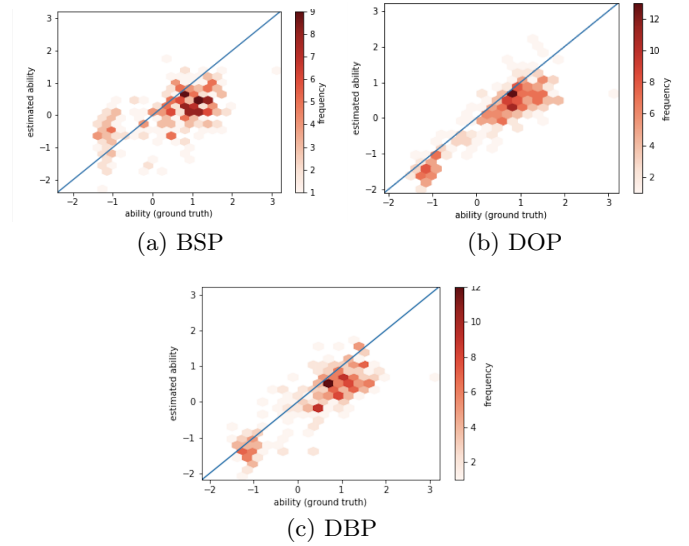


Figure 6: Scatterplots of the abilities of the generated test papers, against the ground truth abilities. Degree of scatter is highest for the BSP paper

Fig. 6 shows a scatterplot of the abilities of the student from the test papers using the three policies, plotted against their ground truth abilities. We can see that the paper generated using BSP policy has the highest degree of scatter and the DOP paper has the lowest, i.e. it most closely matches the ground truth abilities of the student.

6.2 Analysis of the simulated test results

Comparing the policies from the score distribution in the generated test papers, we can see that the DOP and DBP policy give a better spread of scores than BSP, meaning they are better in discriminating between students of different abilities. Tests generated by both DOP and DBP policies also had a higher rank correlation than the BSP test, meaning we get a better accuracy at diagnosing the ability of the students.

The DBP test had a lower spread and lower rank correlation as compared to the DOP test. This could be because we only used the standard 2 PL model of IRT, without any modifications to include behavior parameters. Moreover, behavior parameters, such as time spent not attempting questions, give a more holistic view of how the student performs in a test (such as indicating the confidence level of the student) than simply the academic performance i.e. how many questions the student answered correctly. Perhaps future test papers could be designed in a way that takes into account these factors when computing the student's score.

6.3 Pilot test

To further validate our model, we conducted a pilot study as follows: We selected a group of M students and asked each student to attempt two test papers, using the same online

Table 3: Comparison of pilot test results generated by BSP policy and DBP policy

	BSP	DBP
Number of students	98	99
Number of questions	30	30
Max/Min/Total score	92/0/120	111/-1/120
Score at 95th/ 5th percentile	76/5	85/1
Average score	43.94	42.56

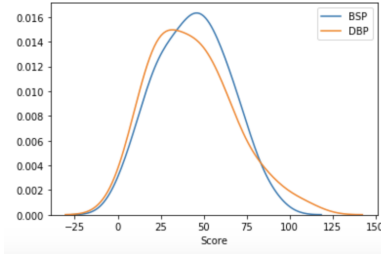


Figure 7: Scores distribution of students in the DBP generated pilot test vs BSP pilot test

learning platform we used for the earlier test generation. For the first paper, we generated the questions using BSP policy and for the second, we generated the questions using DBP policy. We then compared the spreads of scores for these test papers. The results are shown in table 3.

On the pilot test papers generated using the two policies, we found that the DBP test gives 18% higher spread of scores (95th percentile score - 5th percentile score), as compared to the BSP test. The mean squared error for the inferred ability of the students compared to the ground truth ability was 1.08 for the BSP test, and 0.86 on DBP. This is 20% less RMSE for DBP compared to BSP. From the scatterplots in

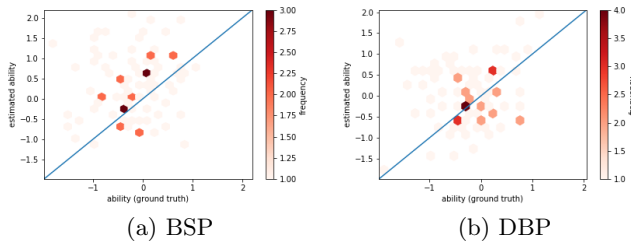


Figure 8: Scatterplots of the inferred abilities of the pilot test papers, against the ground truth abilities

fig. 8 for the inferred vs ground truth ability, we can further confirm that the degree of scatter is lower in the DBP pilot test and higher in the BSP pilot test paper. This confirms that the DBP test paper more accurately reflects the ability of the student, and is also better at discriminating between students of different abilities. The spread of scores in the DBP is better than that of the BSP policy. This validates our findings from the simulated tests, where also we obtained

a better spread for the diagnostic policies (DOP and DBP). Moreover, the higher accuracy of the inferred ability for the DBP pilot test is confirmed by a lower value of the RMSE and lesser degree of scatter compared with BSP.

7. CONCLUSION AND FUTURE WORK

In this paper, we proposed a few policies to generate test papers by selecting a list of questions from a question database. We validated the policies by a pilot test of test papers generated using two policies. We found that the policy of selecting questions based on highest discrimination ability for a given difficulty level yielded the best results.

In future, we intend to extend the IRT model to include behavioral parameters and further validate our method of selecting policies with more candidate policies and a larger sample size of students.

8. ACKNOWLEDGMENTS

The authors express their gratitude to Anwar Sheikh and his team for helping us conduct the pilot study.

9. REFERENCES

- [1] Guang Cen, Yuxiao Dong, Wanlin Gao, Lina Yu, Simon See, Qing Wang, Ying Yang, and Hongbiao Jiang. A implementation of an automatic examination paper generation system. *Mathematical and Computer Modelling*, 51, 2010.
- [2] John Michael Linacre. Computer-adaptive testing: A methodology whose time has come. *Development of computerized middle school achievement test*, 69, 2000.
- [3] Frank B. Baker. *The basics of item response theory*. ERIC, USA, 2001.
- [4] G. Gage Kingsbury. Adaptive item calibration: A process for estimating item parameters within a computerized adaptive test. *GMAC conference on computerized adaptive testing*, 2009.
- [5] Guido Makransky. An automatic online calibration design in adaptive testing. *Journal of Applied Testing Technology*, 11(1), 2014.
- [6] Wim J. van der Linden and Peter J. Pashley. Item selection and ability estimation in adaptive testing. *Elements of adaptive testing*, Springer, 2009.
- [7] Ou Lydia Liu, Brent Bridgeman, and Rachel Adler. Measuring learning outcomes in higher education. *Educational Researcher*, 41, 2012.
- [8] Sideridis GD Tsaousis, I and AA Sadaawi. An irt-multiple indicators multiple causes (mimic) approach as a method of examining item response latency. *Frontiers in psychology*, 9, 2018.
- [9] Ryszard Jaworski. Personalization and calibration of the control question in the control question test. *Journal of Forensic Identification*, 61(5), 2011.
- [10] Shayan Doroudi, Vincent Alevan, and Emma Brunskill. Where’s the reward?: A review of reinforcement learning for instructional sequencing. *International Journal of Artificial Intelligence in Education*, 2019.
- [11] Keras documentation. <https://keras.io>, 2015.
- [12] Ronald K Hambleton and Wim J Linden. *Handbook of modern item response theory. Volume two: Statistical tools*. CRC Press, USA, 2016.

Differential Responses to Personalized Learning Recommendations Revealed by Event-Related Analysis

Kevin C. Dieter

Jamie Studwell

Kirk P. Vanacore

Lexia Learning Systems, LLC, A Rosetta Stone Company

research@lexialearning.com

ABSTRACT

Educators are increasingly embracing personalization in online and blended learning programs as a means of focusing students' investment of time and energy into learning plans that are best tailored to their individual needs. When personalized learning tools are deployed into structured learning environments like schools, however, educators and students must consider program provided recommendations alongside potentially immutable factors like set daily schedules, mandated curricula, and student needs in other content areas. These on-the-ground factors make researching the impacts of personalized learning challenging because they are difficult to measure directly, especially for digital programs deployed at scale. Inspired by a widely influential methodology in brain imaging, we tackled this challenge by employing an *event-related* approach that emphasizes changes in student behavior that are time-locked to changes in program provided usage recommendations. Our analysis reveals that while student usage time can often be quite far from the amount recommended, students nevertheless respond to changes in program recommendations by adjusting usage in a corresponding manner. We further extend this general approach to demonstrate that students more often stayed on track toward their end of year goals following a week where they met or exceeded their program provided recommendation. Through these examples, we demonstrate the value of an event-related approach towards understanding how personalized paths can positively influence student learning.

Keywords

Personalized learning, event-related analysis, time management, K-12 schools, personalized recommendations.

1. INTRODUCTION

As schools and communities embrace a rapidly changing world, a growing emphasis on the personalization of learning has emerged [10]. Learning is considered personalized if it is tailored to each learner's strengths, needs, and interests, encouraging flexibility in a student's pursuit of mastery and enabling learners to take an active role in what, when, where, and how they learn [22]. In the competition for instructional time, personalized learning approaches also hold the promise of helping students achieve mastery as efficiently as possible [10], and can facilitate

educators' work in guiding students' learning efforts towards educational activities that best match their current needs.

Online and blended learning programs are uniquely positioned to enable personalized learning because they can support student agency through independent pacing, delivery of differentiated content and support, and the ability to engage with learning anytime and anywhere [22]. However, the double-edged sword of personalized learning is that "the process of personalization puts enormous pedagogical and procedural burden on the students—as well as teachers—to make critical instructional decisions" [4; also see 5]. This includes decisions about how much time students should spend on specific programs and components of programs to maximize learning. While studies often find that students fail to spend as much time in educational technology programs as recommended by the program or researchers [23], students can also over-use, spending time on one set of activities that might be better spent in other areas.

One response from the designers of learning technologies has been the inclusion of embedded recommendations and self-monitoring tools to scaffold student and teacher support for self-regulation. Recommendations are tailored to help students and teachers make good decisions within a personalized learning environment without enforcing rigid requirements that may reduce student agency and be unrealistic for particular educational contexts. Individualized usage time recommendations do not appear to be common in most learning technologies; many continue to provide one-size-fits-all usage recommendations [9]. However, they hold the promise of facilitating self-pacing by helping students who are at different levels and progressing at different speeds to stay on track toward reaching their goals.

Despite the recognition of learning scaffolds as critical and effective for self-regulation in general [15] and in computer-based learning environments in particular [27, 28], relatively little research has been done into the impacts of recommendations. While the desire to enact personalization grows, the reality is that many educational institutions, particularly K-12 schools, continue to look much as they have for the past century, with set daily schedules and highly-regulated or mandated paths through content material [10]. When individualized learning tools are deployed into schools with structured learning environments, educators and students must consider program provided recommendations alongside these potentially immutable factors. While a program may recommend a different usage time to individual students within the same class or to the same student in different months, they may be unable to follow those recommendations with fidelity because of set schedules of technology access [25], challenges associated with implementing flexible learning time [20], or teacher and parent beliefs about learning technologies and screen time [6, 18]. Furthermore, researchers often have data on the usage recommendations a student received and their time spent

Kevin Dieter, Jamie Studwell and Kirk Vanacore "Differential Responses to Personalized Learning Recommendations Revealed by Event-Related Analysis" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 736 - 742

using the program, but lack direct insight into the specific contexts in which the program is implemented [26].

We addressed this research challenge by examining how students respond to personalized usage time recommendations within *Lexia® Core5® Reading* (hereafter “Core5”) - a blended learning literacy program that provides instruction in foundational literacy skills for students in grades preK-5. To isolate the impacts of program provided recommendations from the largely unknown aspects of school context, we model an analytic approach after one widely used in the field of brain imaging - *event-related design* [for a general overview see 12, 13; for a widely-cited early example see 11]. The key aspect of this methodology with respect to our present application is that we focus on *changes* in actual student usage time that occur time-locked to *changes* in personalized recommendations. That is, we ask how student behavior *responds* to changes in program suggestions rather than whether it is *aligned* to recommendations, as consistent student responses to changed program recommendations could be observed even if baseline student usage is widely variable across diverse school contexts. Utilizing this approach, we find that students indeed adjust the amount of time per week that they spend using Core5 in a manner that differentially relates to the direction and magnitude of the recommended change. We also extend our analysis to examine events defined directly by student behavior and find that the act of meeting one’s recommendation in a given week is associated with more frequently staying on track toward end-of-year goals in future weeks. Together, these examples highlight the power of an *event-related* approach, and reveal positive associations between the personalization of learning and student progress within school contexts.

2. DATA

2.1 Usage time recommendations in Core5

To personalize the learning path for each individual student, Core5 recommends a number of minutes per week that the student should use the online portion of the program, promoting regular, right-sized use and proactive time management throughout the school year to enable student success [7]. Each student’s usage recommendation reflects the estimated amount of time needed to reach their end of year “benchmark” - that is, to complete all program content for their grade level by the end of the school year. It is based on a predictive model that takes into account a student’s current place in the program, the amount of material left for them to reach their benchmark, and their time spent and progress made in the prior month [16]. These recommendations are shared prominently with educators in the program’s online data portal, and are visible to students while logged in to Core5.

Critically, student usage recommendations are not fixed throughout the school year, but are recalculated at the start of each month to reflect student progress and pace (see Figure 1). At the start of the year, before enough data has been collected to personalize recommendations, all students are set to a default recommendation of 40 minutes per week. At the start of the next calendar month (first Monday), a student’s recommendation changes to 20, 30, 50, or 60 minutes per week. With the beginning of each new month, a student’s usage recommendation is recalculated, resulting in either an additional change or a static recommendation. This cadence was chosen to allow regular revisions that reflect student’s usage and progress, while still remaining implementable for teachers. The goal in personalizing and updating these recommendations is that students use the

program enough to stay on pace to end the year at their grade level benchmark, without spending more time than necessary that could be invested in other learning activities. Previous research has shown that students who consistently meet recommended usage in Core5 make more progress and more often reach their grade-level benchmark than those students who infrequently or never meet their usage recommendation [17].

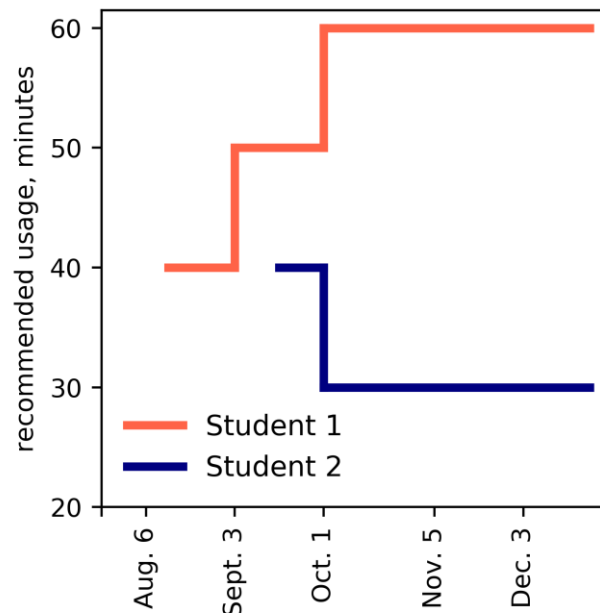


Figure 1: Usage recommendation profiles for two example students. Each line illustrates how usage recommendations for an individual student change across the time frame under study. Student 1 began the school year in early August, and like all students was initially recommended 40 minutes per week. On the first Monday of September, Student 1’s recommendation changed to 50 minutes per week, and at the start of October it was adjusted again to 60 minutes per week. Student 2 was also initially set at 40 minutes per week, but began the school year later (in mid-September). Following the same rules, however, Student 2’s recommendation was adjusted at the start of the next calendar month (October) to 30 minutes per week. For this student, the recommendation remained there for the duration of this time frame.

2.2 Sample Details

Weekly usage records for Core5 students in Kindergarten through 3rd grade were used for the analyses presented in this paper. Although Core5 also provides usage recommendations for pre-K, 4th, and 5th grade students, the specific time values differ for these grade levels. We therefore restricted our sample to K-3 students for clarity of interpretation, though we anticipate that results would be similar for students in other grades.

To obtain the records, schools were chosen at random from among those who had at least one student using Core5 in the fall of 2018 (total of 168 schools chosen). These schools were geographically diverse, located across 39 US states and 4 Canadian Provinces. Student-level demographic data is unavailable for this dataset. All weekly Core5 usage records between August 6, 2018 and December 31, 2018 were obtained for all students at these schools. To be included in the final

sample, students must have used Core5 for at least 7 weeks within this date range, and had their usage recommendation change at least once (as described in Section 1, the goal of the event-related approach is to focus on these changes). In addition, students who met their end of year benchmark (completed all grade level material) within this timeframe were excluded (this sets ones' usage recommendation to 0 minutes per week). Furthermore, students must have had a usage target of 40 minutes in their first week of program use within this timeframe. As previously described, Core5 assigns a default recommendation of 40 minutes per week during a student's first month of use in a school year, and any other value at that time point is an indication that there was a manual override (this is rare - 0.9% of students in our sample - but an available option for educators). The final sample contained 10,851 students (2,838 in Kindergarten; 3,213 in 1st grade; 2,836 in 2nd grade; 1,964 in 3rd grade). To ensure that these exclusion criteria did not produce non-representative results, we ran robustness checks using different cutoffs for minimum weeks of program use (6 or 8) and repeated our analyses with two additional samples of students based on new random selections of schools. We found that all results were qualitatively consistent with our reported findings.

The weekly Core5 records obtained contain aggregated usage data for each week that a student logged into the program. The metrics collected that are relevant to the presented analyses include the total time of Core5 use during that week, the recommended use time for that week, whether or not a student met their recommendation (total time greater than or equal to recommended time), and the Monday date of the week reported.

3. RESULTS

3.1 Alignment to usage recommendations

Our primary research aim is to assess whether students' usage time is responsive to Core5's personalized recommendations. Before presenting our results, however, it is critical that we distinguish this question of students' *responsiveness* to personalized recommendations from a related question about *alignment* between recommended and actual usage time. Specifically, we could observe changes in actual Core5 usage following a change in the program provided recommendation (i.e. a personalized *response*) without necessarily finding that students used the program for a particular number of minutes that is close to their recommended value (i.e. *alignment*). Indeed, because Core5's personalized recommendations serve as only one factor within the school context, it would not be surprising if a student's usage time in a given week was quite far from their personalized recommendation value, and more closely related to unknown (from a researcher's perspective) contextual factors such as the amount of time dedicated in their school's schedule to literacy learning or student-directed after school usage. Critically, even if there is poor *alignment*, we may find that when recommendations are changed that students' time spent using the program systematically adjusts in a manner consistent with those changes - a result indicative of responsiveness to Core5 recommendations.

We indeed find that alignment between Core5's usage recommendations and actual student usage time is weak. Although most students had a mean weekly usage time that fell within the range of Core5 recommendations (Figure 2, top panel; 70.0% of students with weekly mean between 20 and 60 minutes), there was a small negative correlation between actual and recommended program use time in aggregate (Pearson $r = -0.117$; 95% CIs = -

0.137, -0.098). Honing in on a snapshot of one particular week in our dataset (Figure 2, bottom panel), it is evident both that there is poor alignment to recommendations, and that there is widespread individual variability in usage time. While the average within-student mean for actual and recommended usage time were similar (46.2 and 43.3 minutes per week, respectively) the across-student standard deviations for these metrics were widely disparate (SDs = 22.9 and 11.1 minutes, respectively).

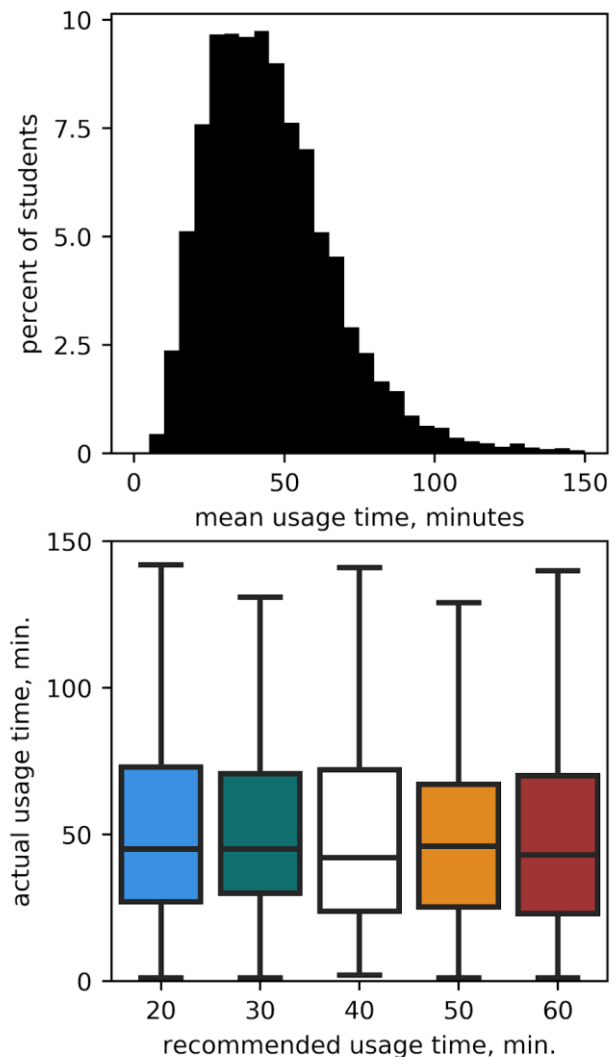


Figure 2: Distributions of students' Core5 usage time. (top) Though mean weekly usage across students was in a range similar to Core5's usage recommendations, there was a notable extent of individual variability. **(bottom)** A snapshot of the distribution of actual use (y-axis) for students with each unique recommended usage time during a particular week (x-axis) revealed no apparent relationship between the two. The data shown is for one example week that had the largest number of unique students using Core5 (week beginning November 26, 2018; 9,606 students, or 88.5% of full sample, had Core5 use), but other weeks had qualitatively similar relationships. The correlation between recommended and actual usage was similar for this sample week (Pearson $r = -0.043$, 95% CIs = -0.063, -0.023) to that seen for the aggregate results.

3.2 Event-related approach to recommendation changes

In light of these observations indicating a lack of positive alignment (Figure 2), we turned to our key question of whether student usage time in Core5 was nevertheless *responsive* to changes in personalized recommendations. Because of the complex and largely unknown (from a researcher’s perspective) context in which these personalized recommendations are implemented, we turned to an *event-related* analytic approach. This methodology allows for context-independent examinations of an event of interest by effectively contrasting responses that occur in temporal coordination with that “event” against a baseline period just before the event occurred [11].

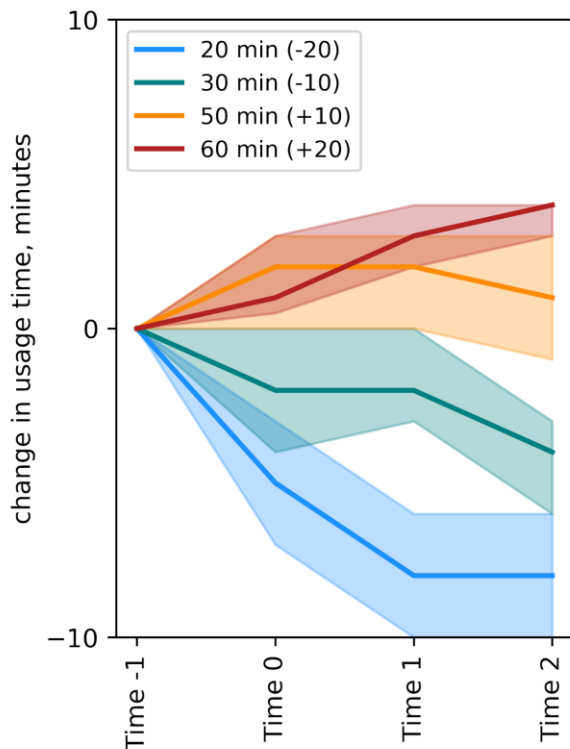


Figure 3: Event-related analysis of usage recommendation changes. When a student’s weekly usage recommendation changes from 40 minutes (Time -1) to another duration (Time 0-2) early in a new school year, student usage tends to change in the direction of (and with magnitude correlated to) the change in recommendation. Each bold line represents the median change in actual weekly usage time across students, with all usage times in the analysis expressed as a difference from Time -1 (this is why all lines converge to 0 at Time -1). Shaded areas around each line represent 95% confidence intervals on the median generated via a bootstrap resampling procedure.

The first step in conducting this analysis is to define the event of interest - here, we focused on how each student’s first change in recommendation influenced their usage time within Core5. Note that because all students began the school year with a recommendation of 40 minutes, this event reflects a change from 40 minutes (at Time -1 in Figure 3) to one of the other four

possible usage recommendations (20, 30, 50, or 60 minutes at Time 0-2 in Figure 3). We next aligned all student data to a temporal reference frame defined by this event. In other words for each student, we defined Time -1 as their last week of program use prior to the recommendation change, Time 0 as the week when the new recommendation first appeared, and Times 1 and 2 as the next two weeks during which that same recommendation remained. Note that these weeks are ordered but are not necessarily consecutive, as students do not always use the program every week. This means it was possible for a student’s recommendation to change *again* at Time 1 or 2 if it fell in the next calendar month. To ensure that the time-course analyzed in Figure 3 reflects the response to the initial target change, we excluded 363 students (3.3%) for whom this occurred, leaving a sample of 10,488. Finally, we subtracted out each student’s actual usage time at Time -1 from all 4 time points to yield a difference metric (this is why all lines converge at 0 for Time -1).

Figure 3 shows the median event-related change in actual student usage time when a recommendation changes from 40 minutes per week at the start of the school year to another value. A two-factor ANOVA (factors of recommendation and time from 0 to 2 in Figure 3, the latter as a repeated measure) revealed significant main effects of both recommendation and time ($F_{3, 10488} = 93.564$, $p < 0.0001$, partial $\eta^2 = 0.0260$; $F_{2, 20968} = 5.71$, $p < 0.0001$, partial $\eta^2 = 0.0005$, respectively), as well as a significant interaction between time and recommendation ($F_{6, 20968} = 10.10$, $p < 0.0001$, partial $\eta^2 = 0.0030$). Repeating this statistical test with a sample that excluded outliers (353 students, or 3.4%, with a change at any time point more than 3 SDs from the mean) produced the same pattern of results. These findings clearly indicate differential responses to Core5 usage recommendations, with decreases in recommended usage (from 40 to 20 or 30 minutes per week) tending to result in decreases in program use, and increases in recommended usage (from 40 to 50 or 60 minutes per week) tending to result in increases. Interestingly, the response to a recommendation change appears to unfold in time, with students continuing to adjust usage time in the direction that their recommendation changed over the next few weeks. This finding further emphasizes the limitations of using snapshot analyses like those in Figure 2 to tease apart effects with unknown temporal dynamics.

It is notable that the median change in program use was smaller in magnitude than the recommended change, especially when one’s Core5 recommendation increased. This observation is consistent with the explanation that contextual factors specific to each student’s school and situation are weighed alongside the program’s personalized recommendations. We also found that despite the visible responses to recommendation changes (Figure 3), that average usage time for all recommendation categories tended to hover around 40 minutes per week (e.g. means in Figure 2, bottom). Such a result suggests a continued reliance on the initially recommended value of 40 minutes per week for all students (see Section 4).

3.3 Event-related approach to student fidelity of program use

As we have demonstrated, taking an event-related approach to studying learning paths in Core5 can clearly reveal differentiated student responses to personalization. While the analysis illustrated in Figure 3 represents one application of this approach to events defined directly by program-driven occurrences (changes made by Core5 at specific points in time), a key advantage of event-related

designs is their flexibility to define new events based on the nature of student actions as well [c.f. 8, 13]. To exemplify this type of approach and to gain more insight into how personalized recommendations influence students' program use, we next define new events based on whether or not a student met or exceeded their recommended usage time in a given week.

Using this definition, we can now ask whether the "event" of a student meeting or exceeding Core5's usage recommendation in a given week is associated with a lasting impact on a student's fidelity of program use, relative to weeks when that same student did not meet her Core5 usage recommendation. In other words, are these helpful recommendations that encourage students to set achievable targets, appropriately pace themselves, and use with fidelity throughout the year [19]? Because it is critical that this analysis be conducted in a within-student fashion (i.e. comparing how the same student responds to both event types), we included only students who had at least one instance of both meeting and not meeting their usage recommendation within the timeframe under study ($N=8,911$; 82.1% of full sample).

Results indicated that students more often met or exceeded their Core5 usage recommendation if they had also met or exceeded their recommendation during their prior week of program use (56.0%, vs. 50.5% when they did not meet or exceed their recommendation during the prior week; odds ratio = 1.248). We also found that while it was very likely overall for students to use Core5 in consecutive weeks, that this was even more frequent following a week of meeting than not meeting one's usage recommendation (88.0% vs. 83.6%, odds ratio = 1.438). Together, these results suggest that following personalized usage recommendations is associated with staying on track toward end of year goals and maintaining regular program use.

4. DISCUSSION

While measuring the impacts of personalized learning in school settings carries significant challenges, we demonstrate the power of an event-related analytic approach toward revealing how student behavior responds to program provided recommendations. Clearly, educators and students must make decisions about personalized recommendations within the context of their school environment and alongside myriad other considerations. The apparent lack of alignment between actual program use and Core5 recommendations (Figure 2), then, is a manifestation of these important but competing priorities. Using an event-related design, we were able to reveal that even within this complex ecosystem, students' Core5 usage time does change in a manner that directly corresponds (and is time-locked) to changes in their personalized recommendations. Furthermore, our results demonstrated that students more often stayed on track toward their end-of-year target following weeks in which they met, versus lagged behind, their suggested pace.

Although usage recommendations are visible to students in the Core5 program, given our sample's age group (K-3) we expect that teachers and school administrators are primarily responsible for monitoring Core5 usage time, responding to recommendations, and weighing program time against other educational priorities. This balancing act likely explains why students' usage time adjustments were typically smaller than was recommended (Figure 3). Together with our other findings, this pattern is consistent with program provided recommendations influencing but not determining student usage time when they are

considered alongside additional factors in each unique school context. In future work it will be interesting to investigate whether responsivity and/or alignment to usage recommendations changes with student age, perhaps reflecting increasing self-regulation and autonomy as they advance in school.

While the ability to isolate one factor of interest from within a complex, dynamical system is a key strength of an event-related approach, it is also a limitation in that it does not afford the ability to quantify influences of other factors or to provide insight as to their relative importance. From the perspective of those designing and improving personalized learning tools, however, an event-related approach is powerful for exactly that reason - it allows for isolated study of a personalized feature that is directly within the designer's control, thus facilitating improvement of the program's design and iteration on these enhancements [c.f. 14]. For example, we noted an interesting finding that even as student usage times changed in response to program provided recommendations, they seemed to remain tied to the initial, impersonal value of 40 minutes per week (e.g. Figure 2, bottom). This pattern may reflect a well-studied cognitive bias known as anchoring, which typically manifests as a continued reliance on an initially given value when making numerical judgments [24]. It may also be that educators have more flexibility to adjust student schedules early in the year than they do as school progresses. In either event, this result suggests that personalizing a student's usage recommendation earlier in the school year could yield larger impacts.

By extending our event-related approach, we found that weeks in which a student met or exceeded their personalized recommendation were more often followed by continued on-track behavior and more regular program use, which have previously been shown to be positive predictors of student performance in online courses [c.f. 7, 21]. Such an effect may stem from integration of Core5's personalized recommendations within educators' learning plans and/or with students' emerging self-regulation [1, 2, 19]. As previously described, our event-related approach limits our ability to quantify effects beyond those owing to program provided recommendations by intentionally filtering them out to isolate only a single factor. That said, these findings motivate further study of the mechanisms through which usage recommendations facilitate students' ability to stay on track for success throughout the school year.

We also note that while our approach is inspired by one developed for the analysis of brain imaging data, it differs in important ways. First and foremost, event-related designs in brain imaging research are typically used in the context of randomized studies, where an experimenter controls many aspects of the timing and context of "events" (although note that the ability to flexibly define events post-hoc is a key methodological advantage, c.f. 8). In contrast, Core5 students are assigned usage recommendations based on their pace through content material and the amount they have left to finish that year. By definition, then, students who are farther behind in class will tend to receive higher Core5 usage recommendations. Although the analyses we present highlight within-student usage changes in response to time-locked events, it is important to note that the groups of students at each recommendation level (e.g. at Times 0-2 in Figure 3) likely differ in other key ways. For example, we may speculate that one reason why usage time increases were typically of smaller magnitude than usage time decreases (Figure 3) could be that students who are farther behind tend to receive offline

interventions at school rather than additional time in the online program.

Analytic applications in the field of brain imaging also suggest extensions of this work that could yield continued insights into the impact of personalization in learning. For example, once well characterized, event-related time courses serve as a template for identifying structural brain regions with particular functional properties [8, 13]. Analogously, having defined the typical time course of how student usage responds to recommendation changes (Figure 3), we could now use these expected functions as regressors to identify schools where recommendations are or are not strongly implemented. This in turn could help guide vendors to better help schools resolve issues and successfully implement digital learning tools. It could also motivate additional research studies that compare student outcomes in school contexts where personalized recommendations either were or were not implemented with fidelity. Such investigations will yield a deeper understanding of the value of personalized recommendations within schools, and in turn provide examples that enable educators to operationalize personalization in their classrooms.

5. ACKNOWLEDGMENTS

The authors thank Lisa B. Hurwitz for helpful feedback on this manuscript.

6. REFERENCES

- [1] Ames, C. 1992. Classrooms: Goals, structures, and student motivation. *J. Educ. Psychol.* 84, 3 (Sep. 1992), 261-271. <https://doi.org/10.1037/0022-0663.84.3.261>.
- [2] Assor, A., Kaplan, H., and Roth, G. 2002. Choice is good, but relevance is excellent: Autonomy-enhancing and suppressing teacher behaviours predicting students' engagement in schoolwork. *Brit. J. Educ. Psychol.* 72, 2 (June 2002), 261-278. <https://doi.org/10.1348/000709902158883>.
- [3] Azevedo, R., and Hadwin, A. F. 2005. Scaffolding self-regulated learning and metacognition – Implications for the design of computer-based scaffolds. *Instr. Sci.* 33, 5 (Nov. 2005), 367-379. <https://doi.org/10.1007/s11251-005-1272-9>.
- [4] Basham, J. D., Hall, T. E., Carter Jr., R. A., and Stahl, W. M. 2016. An operationalized understanding of personalized learning. *Journal of Special Education Technology* 31, 3 (Aug. 2016), 126-136. <https://doi.org/10.1177/0162643416660835>.
- [5] Bingham, A. J., Pane, J. F., Steiner, E. D., and Hamilton, L. S. 2018. Ahead of the curve: Implementation challenges in personalized learning school models. *Educ. Policy* 32, 3 (May 2018), 454-489. <https://doi.org/10.1177/0895904816637688>.
- [6] Blum-Ross, A., and Livingstone, S. 2016. *Families and screen time: current advice and emerging research*. Media Policy Project, London School of Economics and Political Science, London, UK.
- [7] Boroujeni, M. S., Sharma, K., Kidzinski, L., Lucignano, L., and Dillenbourg, P. 2016. How to quantify student's regularity? In *Adaptive and Adaptable Learning*. (EC-TEL 2016). *Lecture Notes in Computer Science*, vol 9891. Springer, Cham. https://doi.org/10.1007/978-3-319-45153-4_21.
- [8] Buckner, R. L. 1998. Event-related fMRI and the hemodynamic response. *Hum. Brain Mapp.* 6, 5-6 (1998), 373-377. [https://doi.org/10.1002/\(SICI\)1097-0193\(1998\)6:5/6<373::AID-HBM8>3.0.CO;2-P](https://doi.org/10.1002/(SICI)1097-0193(1998)6:5/6<373::AID-HBM8>3.0.CO;2-P).
- [9] ETI Consulting. 2018. *Utah's early intervention reading software program: 2018-19 program evaluation findings*. Evaluation and Training Institute, Culver City, CA. <https://www.schools.utah.gov/file/ae750095-378d-4c5e-a7af-1ac0268610b5>
- [10] Friend, B., Patrick, S., Schneider, C., and Vander Ark, T. 2017. *What's possible with personalized learning?* International Association for K-12 Online Learning (iNACOL). Vienna, VA.
- [11] Friston, K. J., Fletcher, P., Josephs, O., Holmes, A., Rugg, M. D., and Turner, R. 1998. Event-related fMRI: characterizing differential responses. *Neuroimage.* 7, 1 (Jan. 1998), 30-40. <https://doi.org/10.1006/nimg.1997.0306>.
- [12] Huettel, S. A. 2012. Event-related fMRI in cognition. *Neuroimage.* 62, 2 (Aug. 2012), 1152-1156. <https://doi.org/10.1016/j.neuroimage.2011.08.113>.
- [13] Huettel, S. A., Song, A. W., and McCarthy, G. 2009. *Functional magnetic resonance imaging* (2nd. Ed.). Sinauer Associates, Inc, Sunderland, MA.
- [14] Kerr, D. 2015. Using data mining results to improve educational video game design. *J. Educ. Data Mining* 7, 3 (June 2015), 1-17. <https://doi.org/10.5281/zenodo.3554723>.
- [15] Kirschner, P. A., Sweller, J., and Clark, R. E. 2006. Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educ. Psychol.* 41, 2 (2006), 75-86. https://doi.org/10.1207/s15326985ep4102_1.
- [16] Lexia Learning Resources. How performance predictors work. (January 2020). Retrieved May 11, 2020 from http://www.lexialearningresources.com/mylexia/Core5_PerfP_reds.pdf
- [17] Lexia Research & Analytics. 2019. *Lexia® Core5® Reading US national progress report: 2018-2019 school year results for approximately 898,000 students*. Lexia Learning Systems, LLC, A Rosetta Stone Company, Concord, MA. <https://www.lexialearning.com/resources/research/lexia-core5-reading-us-national-progress-report-18-19>
- [18] Miranda, H. P., and Russell, M. 2012. Understanding factors associated with teacher-directed student use of technology in elementary classrooms: A structural equation modeling approach. *Br. J. Educ. Technol.* 43, 4 (July 2012), 652-666. <https://doi.org/10.1111/j.1467-8535.2011.01228.x>.
- [19] Niemiec, C. P., and Ryan, R. M. 2009. Autonomy, competence, and relatedness in the classroom. Applying self-determination theory to educational practice. *Theor. Res. Educ.* 7, 2 (June 2009), 133-144. <https://doi.org/10.1177/1477878509104318>.
- [20] Pane, J. F., Steiner, E. D., Baird, M. D., Hamilton, L. S., and Pane, J. D. 2017. *Informing progress: Insights on personalized learning implementation and effects*. RAND Corporation, Santa Monica, CA.
- [21] Park, J., Yu, R., Rodriguez, F., Baker, R., Smyth, P., and Warschauer, M. 2018. Understanding student procrastination

- via mixture models. In *Proceedings of the 11th International Conference on Educational Data Mining (EDM)*, Buffalo, NY, 187-197.
- [22] Patrick, S., Kennedy, K., and Powell, A. 2013. *Mean what you say: defining and integrating personalized, blended and competency education*. International Association for K-12 Online Learning (iNACOL). Vienna, VA.
- [23] Stanhope, D., and Rectanus, K. 2015. *Current realities of EdTech use*. LeaRn. Raleigh, NC.
- [24] Tversky, A., and Kahneman, D. 1974. Judgment under uncertainty: heuristics and biases. *Science* 185, 4157 (Sep. 1974), 1124-1132. <https://doi.org/10.1126/science.185.4157.1124>.
- [25] Vega, V., and Robb, M. B. 2019. *The common sense census: inside the 21st-century classroom*. Common Sense Media. San Francisco, CA.
- [26] Warnakulasooriya, R., and Black, A. 2018. *Beyond the hype of big data in education*. Macmillan Learning. New York, NY.
- [27] Winters, F. I., Greene, J. A., and Costich, C. M. 2008. Self-regulation of learning within computer-based learning environments: a critical analysis. *Educ. Psychol. Rev.* 20 (July 2008), 429-444. <https://doi.org/10.1007/s10648-008-9080-9>.
- [28] Zheng, L. 2016. The effectiveness of self-regulated learning scaffolds on academic performance in computer-based learning environments: a meta-analysis. *Asia Pac. Educ. Rev.* 17 (Apr. 2016), 187-202. <https://doi.org/10.1007/s12564-016-9426-9>.

Prescribing Deep Attentive Score Prediction Attracts Improved Student Engagement

YOUNGNAM LEE^{*1}, BYUNGSOO KIM^{*1}, DONGMIN SHIN¹, JUNGHOO KIM¹,
JINEON BAEK^{1,2}, JINHWAN LEE¹, YOUNGDUCK CHOI^{1,3}

¹Riiid! AI Research, ²University of Michigan, ³Yale University

{yn.lee, byungsoo.kim, dm.shin, junghoon.kim, jineon.baek, jh.lee, youngduck.choi}@riiid.co

ABSTRACT

Intelligent Tutoring Systems (ITSs) have been developed to provide students with personalized learning experiences by adaptively generating learning paths optimized for each individual. Within the vast scope of ITS, score prediction stands out as an area of study that enables students to construct individually realistic goals based on their current position. Via the expected score provided by the ITS, a student can instantaneously compare one's expected score to one's actual score, which directly corresponds to the reliability that the ITS can instill. In other words, refining the precision of predicted scores strictly correlates to the level of confidence that a student may have with an ITS, which will evidently ensue improved student engagement. However, previous studies have solely concentrated on improving the performance of a prediction model, largely lacking focus on the benefits generated by its practical application. In this paper, we demonstrate that the accuracy of the score prediction model deployed in a real-world setting significantly impacts user engagement by providing empirical evidence. To that end, we apply a state-of-the-art deep attentive neural network-based score prediction model to *Santa*, a multi-platform English ITS with approximately 780K users in South Korea that exclusively focuses on the TOEIC (Test of English for International Communications) standardized examinations. We run a controlled A/B test on the ITS with two models, respectively based on collaborative filtering and deep attentive neural networks, to verify whether the more accurate model engenders any student engagement. The results conclude that the attentive model not only induces high student morale (e.g. higher diagnostic test completion ratio, number of questions answered, etc.) but also encourages active engagement (e.g. higher purchase rate, improved total profit, etc.) on *Santa*.

Keywords

Intelligent Tutoring System, Score Prediction, Engagement, Deep Learning, Transformer

YOUNGNAM LEE, BYUNGSOO KIM, DONGMIN SHIN, JUNGHOO KIM, JINEON BAEK, JINHWAN LEE and YOUNGDUCK CHOI "Prescribing Deep Attentive Score Prediction Attracts Improved Student Engagement" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 743 - 748

1. INTRODUCTION

The significance that standardized examinations (e.g. SAT and TOEIC) currently hold is to provide an objective criteria in which each individual's academic performance is measured. Accordingly, Intelligent Tutoring Systems (ITSs), which generate optimized learning paths for each student, often include functions such as estimating expected performance on standardized examinations. In this regard, measuring the expected academic performance of a student has become an interesting area of study in Artificial Intelligence in Education (AIED). These studies focus on modelling a student's understanding of a target subject based on their learning activities. For instance, Matrix Factorization (MF) [10, 22, 16, 17, 23, 7, 25, 24] is a prevalent method used for grade prediction, in which the latent vectors of students and courses are learned by factorizing a student-grade matrix into two low-rank matrices. Markov and semi-Markov models are also some other popular approaches for grade prediction [11, 7, 23]. With the advances in deep learning, neural network based models with deeper hidden layers, such as Multi-Layer Perceptron, Recurrent Neural Networks and Convolutional Neural Networks, were introduced to predict student's academic performance [21, 9, 11, 8]. In [3], the Transformer-based [29] bidirectional encoder model was first pre-trained to predict masked assessments and then fine-tuned to predict exam score, resulting in a state-of-the-art score prediction model. Although precision of academic performance prediction is significant as it is directly associated to a reliability of an ITS, previous studies have mainly focused on improving the accuracy of the prediction, leaving discussion about the benefits of precise prediction on student engagement fairly opaque.

In this paper, we direct our attention towards the correlation between the precision of score prediction and student engagement. Our study starts by hypothesizing that students will show higher level of engagement if they experience a more precise score prediction while interacting with ITS. We empirically verify our hypothesis on *Santa*, a multi-platform English ITS with approximately 780K users in South Korea that exclusively focuses on the TOEIC (Test of English for International Communications) Listening and Reading Test Preparation. In the experimental studies, we run a controlled A/B test with two score prediction models that differ in accuracy, which are respectively based on collab-

*Equal contribution.

orative filtering with Mean Absolute Error (MAE) of 78.9 and deep attentive neural networks with MAE of 49.8. The results show that the superior performing, deep attentive neural network based score prediction model induces more student engagement. These benefits range from ones that are derived from learning behavior (e.g. preliminary test completion ratio, membership rates, the average number of questions a student answered after the diagnostic test) to more active engagement (e.g. purchase rate, average revenue per user, and total profit). To the best of our knowledge, this is the first work studying the benefits of accurate score prediction of ITS on student engagement.

2. RELATED WORKS

The related works of this study can be grouped into two categories: academic performance prediction and student engagement.

2.1 Academic Performance Prediction

Predicting a student's academic performance is a significant aspect in solving the problems within AIED. A successful prediction model can be used to recommend appropriate courses, provide interventions for at-risk students, and optimally allocate learning materials. Extensive work has been conducted on performance prediction, exploring a wide range of methodologies from simple regressions to deep learning.

The most widely used methodology in grade prediction is low rank Matrix Factorization (MF) [10, 22, 16, 17, 23, 7, 25, 24]. Low rank MF assumes that there is a low-dimensional latent space containing features that can effectively represent both students and the academic tasks students will be graded on. These features can be interpreted as representations of a student's knowledge. We find these features by decomposing a student-grade matrix into a product of two low-rank matrices. The authors of [22] show that the MF-based model outperforms other course/student-specific regression models. [16] improved the model by assuming that different courses share a common latent feature space, since the totality of a student's knowledge should not change based on the courses they are taking.

Markov and semi-Markov models are another popular set of models for grade prediction [11, 7, 23]. These models capture the dynamic evolution of a student's learning status and leverage it to effectively predict outcomes. [7] develops course-specific hidden Markov and semi-Markov models for grade prediction. [11] models student behavior in MOOCs by using Hidden Markov Models (HMMs) and Multinomial Mixture Models (MMMs) to cluster sequences of student actions. The study applies an LSTM model to predict the students' final grades. Markov models are also used to estimate a student's performance on educational games [28] or to predict student retention in MOOCs [1].

[21, 9, 11, 8] introduce deep-learning based prediction models. The authors of [9] introduce two types of Bayesian deep learning models for grade prediction using Multi-Layer Perceptron and LSTM architectures. Their results show that their model outperforms several baseline models (including MF-based models and course-specific regression models) in detecting at-risk students. The authors of [3] propose As-

essment Modeling (AM), a pre-training method applicable to general ITSs. In AM, a model is first pre-trained to predict several assessments of a student automatically made by ITS during one's learning process. Their results show that a Transformer [29] based neural network model with AM improves model accuracy compared to the same network with other state-of-the-art pre-training methods (such as BERT [5] based word embedding and QuesNet [31] question embedding) on exam score prediction and review correctness prediction.

2.2 Student Engagement

Student engagement is also an actively studied topic in the field of AIED. Several works have analyzed student engagement patterns to figure out which factors vastly impact engagement. [30] studied how people use digital textbooks and compare engagement patterns among high school students, college students, and online website viewers. [18] investigated student engagement in an online learning system which outperformed a traditional classroom on key indicators of engagement, such as time on-task, engaged concentration, and boredom. [26] found correlations between semantic features of mathematics problems and indicators of engagement. [14] discriminated behavioral engagement and cognitive engagement, and argued that most of students who were behaviorally engaged were not cognitively engaged.

Another line of student engagement research focused on predicting engagement level. [20] proposed a two-phased approach for automatic engagement detection, which utilized contextual logs and appearance information to infer behavioral engagement. [19] investigated the relationship between engagement and performance. Firstly, this work analyzed log traces for each learner to calculate engagement indicators that represent learner's engagement level. Based on the quantified engagement indicators, prediction on the learner's performance were attempted.

Enrollment is a sign of strong engagement since it involves determination that a student must invest to take a certain course. Accordingly, predicting and promoting enrollment is highly relevant to student engagement research. [6] proposed a novel extension of Factorization Machines to infer students' course enrollment information from incomplete data. [2] presented a course enrollment recommender system which recommended selective and optional courses based on students' skills, knowledge and interests. [27] identified factors that affect the likelihood of enrolling. This work analyzed the enrollment predictability of such factors using logistic regression, support vector machines, and semi-supervised probability methods.

With the development of Massive Open Online Courses (MOOCs), several works studied student engagement in a MOOCs environment. [12] proposed a recommender which provides each student with an individual list of contacts based on their own profile and activities to foster their engagement in MOOCs. [15] investigated the relationship between students' self-evaluation of their previous knowledge and students' engagement behaviors in MOOCs through a polytomous item response theory model.

3. SCORE PREDICTION MODELS

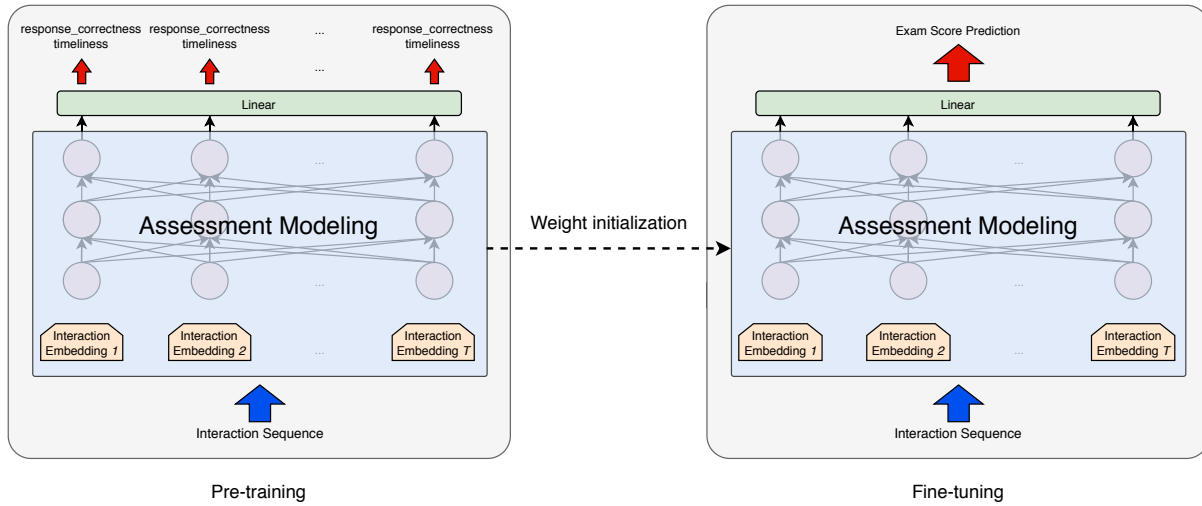


Figure 1: Pre-training/fine-tuning scheme of Assessment Modeling for score prediction. First, a model is pre-trained to predict two assessments: response correctness and timeliness. After pre-training, the last layer of the model is replaced with a layer with randomly initialized weights and appropriate dimension for score prediction. The parameters in the model are fine-tuned to predict exam scores.

Our studies are based on comparing the two approaches for score prediction: a collaborative filtering based approach and Assessment Modeling. The following subsections briefly cover each approach. More detailed descriptions can be found in [13] and [3].

3.1 Collaborative Filtering based Approach

There are two phases in the Collaborative Filtering (CF)-based score prediction approach. First, the CF-based model developed in [13] estimates the probability that a student responds correctly to each potential question. In this model, each user or question is represented as a k -dimensional latent vector, where k is the number of hidden concepts. For instance, if there are n users with m questions, we have user vectors L_1, L_2, \dots, L_n and question vectors R_1, R_2, \dots, R_m each with dimension k . The knowledge level of user i understanding question j is represented as $X_{ij} = L_i \cdot R_j$. Accordingly, the probability of user i getting question j correct is modeled as

$$\phi(X_{ij}) = \phi_a + \frac{1 - \phi_a}{1 + e^{-\phi_c(X_{ij} - \phi_b)}},$$

where ϕ_a , ϕ_b , and ϕ_c are parameters appropriately set, independently of questions or users. The learning algorithm then finds the maximum likelihood estimator by minimizing the negative of log-likelihood of observed user-question entries with Frobenius norm regularizer terms through the projected stochastic gradient descent.

Given the response correctness probabilities calculated from the CF-based model, scores for Listening Comprehension (LC) and Reading Comprehension (RC) are calculated through the following quadratic equations

$$\begin{aligned} score_{LC} &= \theta_2 x_{LC}^2 + \theta_1 x_{LC} + \theta_0 \\ score_{RC} &= \theta_5 x_{RC}^2 + \theta_4 x_{RC} + \theta_3, \end{aligned}$$

where x_{LC} and x_{RC} are each the average of predicted response correctness probability of potential questions in LC and RC, and θ s are properly set parameters. The final score is the sum of $score_{LC}$ and $score_{RC}$.

3.2 Assessment Modeling

[3] introduced Assessment Modeling (AM), a fundamental pre-training method for general class of ITSs. The motivation behind the works of AM is to deal with label-scarce problems in AIED. Score prediction is a typical example of such label-scarce educational problems since standardized exam scores are not obtainable within ITS. Collecting the exam scores involves student action taken outside ITS. The approach proposed in [3] is based on a pre-training/fine-tuning paradigm. In the pre-training phase, the Transformer-based [29] bidirectional encoder model is trained to predict randomly masked assessments, which are interactive educational features available in ITS. Examples of these assessments include response correctness (whether a student provides a correct response to a given question) and timeliness (Whether a student responds to each question within the time limit specified by domain experts). In the fine-tuning phase, the last layer of the pre-trained model is replaced with a randomly initialized layer with an appropriate dimension for a specific downstream task. Afterwards, the parameters in the model are updated to predict labels in the downstream task. In the experimental studies conducted on EdNet [4], AM outperformed pre-training methods that learn the contents of learning materials in several downstream tasks including score prediction. See Figure 1 for graphical description of AM.

4. EXPERIMENTS

4.1 Santa service

Santa is a multi-platform English ITS with approximately 780K users in South Korea that exclusively focuses on the

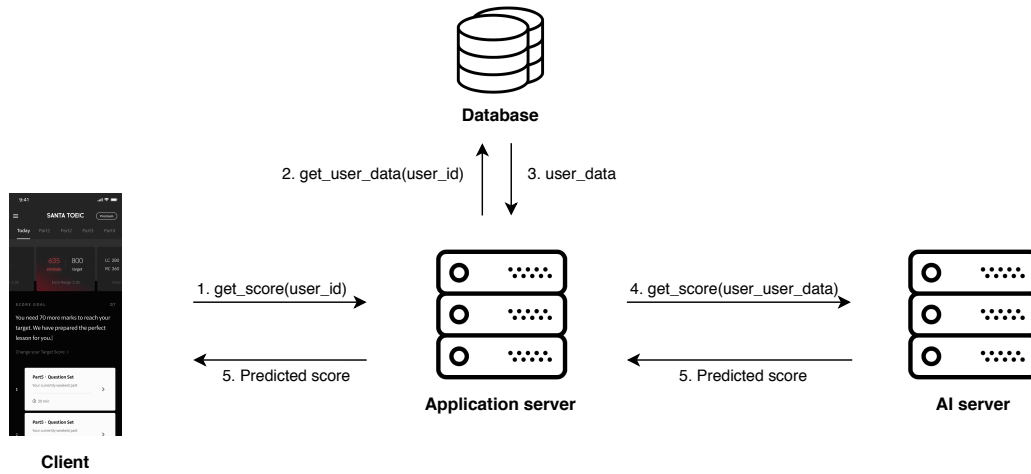


Figure 2: The flow of score prediction.

TOEIC (Test of English for International Communications) standardized examinations. TOEIC is an English proficiency test that consists of two timed sections (listening and reading) each with 100 questions that adds up to a combined total score between 0 to 990. Santa utilizes several AI techniques to optimize the preparation process of the TOEIC examination for students. When the application is first initiated, a preliminary placement test with 7 to 11 problems is given to diagnose the student’s current state and predict their expected score in real-time. After the diagnostic test, a user response prediction model is used to dynamically suggest problems which corresponds to the student’s current position within the TOEIC ladder. The prediction model is calculated by computing a user’s overall correctness rate, eliminating problems that students have answered correctly with high probability and then selecting the best possible content based on expert heuristics. Based on the user’s previous data, the predicted scores can be provided in various forms throughout the service, as shown in Figure 3. Figure 2 shows the flow of score prediction.

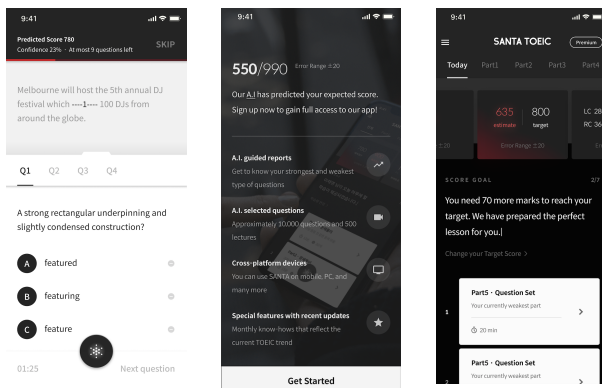


Figure 3: Various representations of predicted scores within the application.

4.2 Performance of Score Prediction Model

Santa has previously used a CF-based model for score prediction which has recently been replaced with a deep attentive model. To train the model, we aggregate the real TOEIC scores reported by users of Santa. Santa offered to reward to the users who have reported their score and was able to obtain a total of 2,594 score reports for 6 months. The data is then divided into a training set (1,302 users, 1815 labels), validation set (244 users, 260 labels), and a test set (466 users, 519 labels). We use EdNet as pre-training task data and the student sequence data as the label (TOEIC score). Table 4.2 shows the MAE (Mean Absolute Error) of the two models for the test set.

	CF	Deep Attentive model
MAE	78.91	49.84

Table 1: MAE of collaborative filtering and attentive model

4.3 A/B test setup

From February 24th to April 2nd, we conducted an A/B test by randomly administering two different score prediction algorithms to the application users: one based on a collaborative-filtering algorithm and another one based on deep-learning. 50,451 students were allocated to the collaborative filtering algorithm and 17,019 students were provided a deep-learning algorithm. We analyzed each student’s response and action (such as time of registration, question response time, purchase rate, etc.) to spot any noteworthy statistics that can validate our experiment.

4.4 Experimental Results

In this section, we discuss how a high quality of the predicted scores can significantly impact student morale.

4.4.1 Student Motivation

Our first test statistic is the preliminary test completion ratio. The completion rate of the initial placement test is a crucial indicator that could represent a student’s motivation,

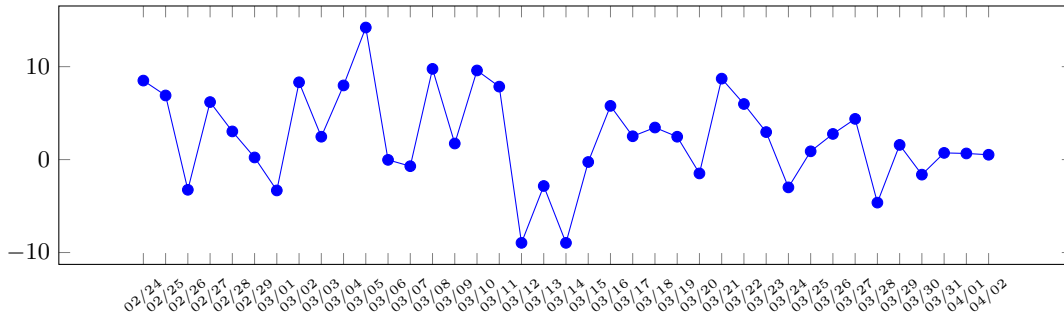


Figure 4: Comparison of the number of questions solved per day between the users of the A/B test.

as only students who are willing to learn will try to finish their diagnostic test. For each question a student answers in SANTA, a predicted score that is updated in real-time is projected on the top left corner. This allows for the user to immediately check the quality of the expected score, thus strengthening the trust that the user may have with the application. A/B test results show that the deep attentive model has a higher completion rate of 64.93% than the CF-model with 65.90%.

Next, we look at changes in membership rates. A membership rate of an application in a sense signifies greater magnitude of student motivation than the completion rate as it directly indicates the determination of a user who wishes to use the application. Out of a total of 67,470 users that have used Santa during the A/B test period, 44,297 users finished their diagnostic tests and 28,065 users have registered to sign up with the application. The A/B test shows that the deep attentive model has a registration rate of 43.13% while the CF-based model has 44.55%.

The average number of questions a user answered after the diagnostic test is also significant proof of a student's educational drive. The A/B test results show that with a deep attentive model a student solved an average of 22.73 questions, while with a CF-based model the user only solved 20.03. Figure 4 shows the comparison of the number of questions answered per day between the users of the A/B test. The x-axis represents the date and the y-axis represents the gap between average number of questions answered in a deep attentive model and a CF-based model. If the gap is positive, the former model has on average more questions solved, and vice versa. We can observe that more questions from the deep attentive model were solved mostly throughout the A/B test time period.

	CF	Deep Attentive model
Completion rate (%)	64.93	65.90
Registration rate (%)	43.13	44.55
# of solved questions	20.03	22.73

Table 2: Experimental results of student motivation

4.4.2 Active Student Engagement

In this section, we demonstrate active student engagement based on different score prediction models via taking a look at the financial benefits the models bring. Monetary profits are an essential factor in evaluating a service, since it is

an important indicator of user engagement as a high level of user engagement directly results in financial success. We measure business impact with 3 metrics : purchase rate, Average Revenue Per User (ARPU), and total profit. In this context, purchase rate is defined as the number of users that decided to purchase full access to the app during the A/B test period. The test results show that the purchase rate for the deep attentive model was 2.73% while the CF-based model had a 2.37% rate, showing a 15.19% increase for the deep attentive model. For ARPU, the deep attentive model averaged \$3.23 whilst a CF-based model averaged \$2.83. Total profit during testing period also yielded \$162,933.88 for the former while it only gathered \$142,949.55 for the latter (since the two models had different parameters, these values were normalized based on the ratio of the model parameters). Comparing these 3 metrics, we conclude that the model with higher accuracy in the deep attentive model shows better results as well.

	CF	Deep Attentive model
Conversion rate (%)	2.37	2.73
ARPU (\$)	2.83	3.23
Total profit (\$)	142,949.55	162,933.88

Table 3: Experimental results of student engagement

5. CONCLUSIONS

Recent developments in ITS have enabled customized education by suggesting optimal strategies for individual students to approach studying. SANTA has also assisted its users to better prepare for the TOEIC English fluency standardized examinations by utilizing various learning techniques. Recently, SANTA has shifted from a collaborative-filtering model to a deep attentive model that has proved to be an upgrade over the former. To inquire about the benefits of using a fastidious model, this paper conducts various experiments and investigates their results. Analyzing the results of various experiments leads us to believe that deep attentive model entails a higher level of student motivation and engagement. Therefore, we claim that a more accurate model, in this case, the deep attentive model, could induce improved student engagement.

6. REFERENCES

- [1] G. Balakrishnan and D. Coetzee. Predicting student retention in massive open online courses using hidden markov models. *Electrical Engineering and Computer*

Sciences University of California at Berkeley, 53:57–58, 2013.

- [2] H. Bydžovská. Course enrollment recommender system. *International Educational Data Mining Society*, 2016.
- [3] Y. Choi, Y. Lee, J. Cho, J. Baek, D. Shin, S. Lee, Y. Cha, B. Kim, and J. Heo. Assessment modeling: Fundamental pre-training tasks for interactive educational systems, 2020.
- [4] Y. Choi, Y. Lee, D. Shin, J. Cho, S. Park, S. Lee, J. Baek, B. Kim, and Y. Jang. Ednet: A large-scale hierarchical dataset in education, 2019.
- [5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [6] J. P. González-Brenes and R. Edezhath. Inferring course enrollment from partial data. In *International Conference on Artificial Intelligence in Education*, pages 429–432. Springer, 2018.
- [7] Q. Hu and H. Rangwala. Course-specific markovian models for grade prediction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 29–41. Springer, 2018.
- [8] Q. Hu and H. Rangwala. Academic performance estimation with attention-based graph convolutional networks. *arXiv preprint arXiv:2001.00632*, 2019.
- [9] Q. Hu and H. Rangwala. Reliable deep grade prediction with uncertainty estimation. *arXiv preprint arXiv:1902.10213*, 2019.
- [10] Z. Iqbal, J. Qadir, A. N. Mian, and F. Kamiran. Machine learning based student grade prediction: A case study. *arXiv preprint arXiv:1708.08744*, 2017.
- [11] Y. Jo, K. Maki, and G. Tomar. Time series analysis of clickstream logs from online courses. *arXiv preprint arXiv:1809.04177*, 2018.
- [12] H. Labarthe, F. Bouchet, R. Bachelet, and K. Yacef. Does a peer recommender foster students’ engagement in moocs?. *International Educational Data Mining Society*, 2016.
- [13] K. Lee, J. Chung, Y. Cha, and C. Suh. Machine learning approaches for learning analytics: Collaborative filtering or regression with experts? In *NIPS Workshop, Dec*, pages 1–11, 2016.
- [14] Q. Li and R. Baker. Understanding engagement in moocs. In *EDM*, pages 605–606, 2016.
- [15] J. Lui and H. Li. Exploring the relationship between student pre-knowledge and engagement in moocs using polytomous irt. In *Proceedings of the 10th International Conference on Educational Data Mining*, pages 410–411. ERIC, 2017.
- [16] S. Morsy and G. Karypis. Cumulative knowledge-based regression models for next-term grade prediction. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 552–560. SIAM, 2017.
- [17] S. Morsy and G. Karypis. Sparse neural attentive knowledge-based models for grade prediction. *arXiv preprint arXiv:1904.11858*, 2019.
- [18] K. Mulqueeney, L. A. Mingle, V. Kostyuk, R. S. Baker, and J. Ocumpaugh. Improving engagement in an e-learning environment. In *International Conference on Artificial Intelligence in Education*, pages 730–733. Springer, 2015.
- [19] V. Naik and V. Kamat. Analyzing engagement in an on-line session. In *International Conference on Artificial Intelligence in Education*, pages 359–364. Springer, 2019.
- [20] E. Okur, N. Alyuz, S. Aslan, U. Genc, C. Tanriover, and A. A. Esme. Behavioral engagement detection of students in the wild. In *International Conference on Artificial Intelligence in Education*, pages 250–261. Springer, 2017.
- [21] A. P. Patil, K. Ganesan, and A. Kanavalli. Effective deep learning model to predict student grade point averages. In *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–6. IEEE, 2017.
- [22] A. Polyzou and G. Karypis. Grade prediction with course and student specific models. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 89–101. Springer, 2016.
- [23] L. Rechkoski, V. V. Ajanovski, and M. Mihova. Evaluation of grade prediction using model-based collaborative filtering methods. In *2018 IEEE Global Engineering Education Conference (EDUCON)*, pages 1096–1103. IEEE, 2018.
- [24] Z. Ren, X. Ning, A. Lan, and H. Rangwala. Grade prediction based on cumulative knowledge and co-taken courses. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM)*. ERIC, 2019.
- [25] Z. Ren, X. Ning, and H. Rangwala. Ale: Additive latent effect models for grade prediction. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 477–485. SIAM, 2018.
- [26] S. Slater, R. Baker, J. Ocumpaugh, P. Inventado, P. Scupelli, and N. Heffernan. Semantic features of math problems: Relationships to student learning and engagement. *International Educational Data Mining Society*, 2016.
- [27] A. Slim, D. Hush, T. Ojah, and T. Babbitt. Predicting student enrollment based on student and college characteristics. *International Educational Data Mining Society*, 2018.
- [28] M. Tadayon and G. Pottie. Predicting student performance in an educational game using a hidden markov model. *arXiv preprint arXiv:1904.11857*, 2019.
- [29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [30] J. Warner, J. Doorenbos, B. Miller, and P. J. Guo. How high school, college, and online students differentially engage with an interactive digital textbook. In *EDM*, pages 528–531, 2015.
- [31] Y. Yin, Q. Liu, Z. Huang, E. Chen, W. Tong, S. Wang, and Y. Su. Quesnet: A unified representation for heterogeneous test questions. *arXiv preprint arXiv:1905.10949*, 2019.

The Results of Implementing Zone of Proximal Development on Learning Outcomes *

Ryan Baker
University of Pennsylvania
ryanshaunbaker@gmail.com

Wei Ma
Renmin University of China
mawei@ruc.edu.cn

Yuxin Zhao
Learnta Inc
zhaoyuxin@learnta.com

Shengni Wang
Learnta Inc
wangshengni@learnta.com

Zhenjun Ma
Learnta Inc
will@learnta.com

ABSTRACT

With the development of personalized learning in technological platforms, more data and information are given to instructors on what contents are appropriate for a learner's next step, with an aim of helping them support their students in navigating an optimized learning path that can promote an enhanced learning outcome. In this study, we collected data from an online learning platform, Learnta[®] TAD, which allows teachers to distribute tasks based on system recommendations. The recommendations are directed by the system's knowledge graph algorithm, determining whether the student is ready to learn the task (i.e. the task is within the student's Zone of Proximal Development), whether the student is not yet ready to learn the task, or whether the student has already mastered the task. We used the acquired data to investigate whether giving content in each of these groups results in different learning outcomes. Statistical methods such as subgroup analysis, Fisher's exact test, and logistic regression are conducted to address the proposed topic. Replicating a prior, smaller-scale study, our findings suggest that the student gains more mastery when assigned Ready-to-Learn tasks than when assigned Unready-to-Learn tasks, across Math and English, more and less successful students, and in-class and homework. Moreover, students who are given already mastered tasks perform better than those who are given Ready-to-Learn and Unready-to-Learn tasks across all groups.

Keywords

Zone of Proximal Development · Knowledge Graph · Ready-to-Learn · Unready-to-Learn

*(Does NOT produce the permission block, copyright information nor page numbering). For use with ACM_PROC_ARTICLE-SP.CLS. Supported by ACM.

Shengni Wang, Yuxin Zhao, Wei Ma, Zhenjun Ma and Ryan Baker "The Results of Zone of Proximal Development on Learning Outcome" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 749 - 753

1. INTRODUCTION

Increasingly, teachers' decisions are driven by data [3], with increasing data becoming available from online learning environments [9]. Using reports from online learning systems, educators are able to track and evaluate each student's learning based on data [1]. However, even though data are given to teachers by these systems, instructors are still impeded by having insufficient knowledge about how to use the data [7]. In other words, teachers still have difficulties in using data effectively to decide what students need to learn next, to maximize learning outcomes and expedite the learning process.

This problem is exacerbated in online learning systems that give relatively more agency to teachers in choosing which content their students will work with. Although such systems are easier to integrate with existing pedagogical practices, they raise questions as to whether teachers will assign the best possible content. We can consider this decision in terms of whether a teacher selects content that falls within a learner's zone of proximal development (ZPD) [8]. A task within a learner's ZPD is one that he or she can succeed in, but only with external support or scaffolding. Tasks that a learner can succeed in without support, and tasks that a learner cannot succeed in even with support, fall outside of the learner's ZPD. Although the ZPD has been a popular concept in the educational literature for decades, only limited attention has been paid to ZPD in educational data mining and related communities [4].

However, recent research has found evidence that Vygotsky's concept of the ZPD can be beneficial to the design of adaptive learning systems [10]. In that work, Zou and colleagues investigated whether teachers make good instructional decisions based on student performance data. They compared "Ready-to-Learn" (RtL) content inside the ZPD to content that students were "Unready-to-Learn" (UtL), using automated assessments of student progress through a curriculum based on a knowledge graph.

We replicate and build on this work with a larger student sample, assessing whether a task is RtL for a specific student using the prerequisite structure within a knowledge graph. Our hypothesis is that, like in [10], students will gain more mastery (successfully complete more objectives within the

system) if they are assigned RtL tasks instead of UtL tasks. We also investigate whether the findings in [10] are robust to whether the student is completing tasks as homework as opposed to in class. We hypothesize that students working on content in class will gain more mastery than students completing tasks as homework, due to the availability of greater learning support and scaffolding in an in-class context [6]. We also investigate whether the findings in [10] are robust to the general level of success of the student. If some students are simply faster or better learners than others in a domain (e.g. [2]), then they may be able to perform better even when given UtL. However, one could also argue that if the knowledge graph is correct, then all students should have similar (poorer) outcomes for UtL content, since regardless of their general ability they lack the building blocks to acquire the content they are given. Finally, we investigate whether the results in [10] are robust across two different learning subjects, English and Mathematics.

2. THE ONLINE LEARNING PLATFORM

The system used in this study is a learning platform for K-12 students in China, called Learnta[®] TAD, developed by Learnta Inc.. Learnta[®] TAD, an acronym of “Teacher + Artificial Intelligence + Data”, is a system which gives teachers data on student learning progress and makes recommendations on optimal learning path using AI algorithms, and then allows teachers to decide which content students should work on. Learnta[®] TAD is primarily used in blended learning, where teachers give students face-to-face instructions in classroom.



Figure 1: Teacher’s Interface of Learnta[®] TAD system

In TAD, teachers assign learning tasks that contain several target skills to the students. The system infers each student’s mastery of each skill using Bayesian Knowledge Tracing (BKT) (Corbett & Anderson, 1995) by predicting the student’s latent knowledge state according to the student’s correctness on questions related to the skills. Learnta’s directed knowledge graph maps content to a prerequisite structure, representing which prerequisite content is necessary to know to learn a particular piece of content. Based on the mastery of the student and the prerequisite structure of each

skill, Learnta[®] TAD recommends RtL contents for teachers to instruct. More specifically, content is considered RtL if the student has mastered all the prerequisites of that skill; UtL indicates that the student is missing one or more of a skill’s prerequisites. Whether or not the teachers choose to follow the recommendations, the system collects data on the students’ performance and learning outcomes. Teachers can assign material that is RtL, UtL, or even Already Mastered (AM).



Figure 2: Teacher using Learnta[®] TAD in classroom

3. DATA COLLECTION

To investigate our research questions around ZPD status and students’ learning outcomes, we collected data from 7913 middle school and elementary school students who studied 250,783 task cards (one task card contains several skills) in Learnta[®] TAD, during 2019.

In the context of both English and Math, we categorized students into different levels based on their earlier assessment test performance: 1) Excellent students; 2) Normal students; 3) Struggling students. Excellent students are those who mastered at least 80% of the skills in the assessment, according to Bayesian Knowledge Tracing. Normal students are those who mastered at least 60% but less than 80% of the skills in the assessment. Struggling students are those who mastered less than 60% of the skills in the assessment. The proportion of these three student categories is 32.39%, 53.78% and 13.83%, respectively.

In addition to that, we compare the use of the system in a classroom setting to its use as homework. In-class, students complete the assigned tasks under the supervision of their teachers during a class session. Within the homework context, students are expected to complete their tasks at home. The percentage of these two scenarios are 57.5 % and 42.5 %, respectively.

4. STATISTICAL ANALYSIS

We compare the learning outcomes of teachers’ decisions of what skills the student should work on. The analyses are conducted on two topics - Math and English - separately. The outcome of interest is whether the student mastered the skill according to BKT. The percentage of skills that are mastered are tabulated for each type of teaching decisions: RtL, UtL, and AM.

In addition to descriptive statistics, we conduct Fisher’s exact test to assess the association between instructional decisions and student mastery. Our hypothesis is that students are more likely to master RtL skills than UtL skills.

In addition, a logistic regression model is used, with learning outcome as the independent variable, and teacher’s decision, student’s level, and whether learning occurs in a classroom as predictors.

P values are calculated in R version 3.6.3 using the `fisher.test()` function for Fisher’s exact test and the `glm()` function for logistic regression.

5. RESULTS

For the tasks in Math, the completion rates were 76.5%, 70%, and 65%, respectively, for the excellent, normal and struggling students. The completion rates were 79%, 72%, and 66.5% in English. Those findings indicate that the students’ completion rates vary depending on overall student success, $\chi^2(df = 2, N = 93874) = 650.29, p < 0.001$ for Math and $\chi^2(df = 2, N = 146127) = 1465.87, p < 0.001$ for English.

The completion rates for in-class tasks were 75.7% for Math and 74.7% for English, and for homework tasks the completion rate were 65.3% for Math, and 70.3% for English (see Figures 3 and 4). Fisher’s exact tests show the in-class tasks were more likely to be completed than the homework tasks for both Math ($p < 0.001$) and English ($p < 0.001$).

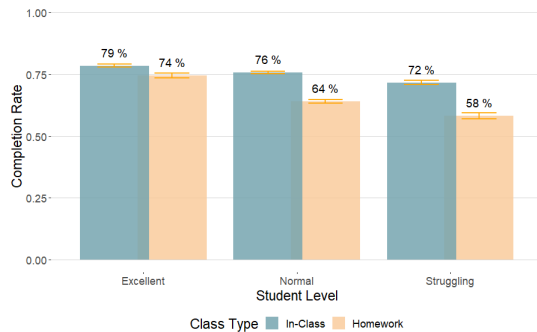


Figure 3: Completion Rate in Math

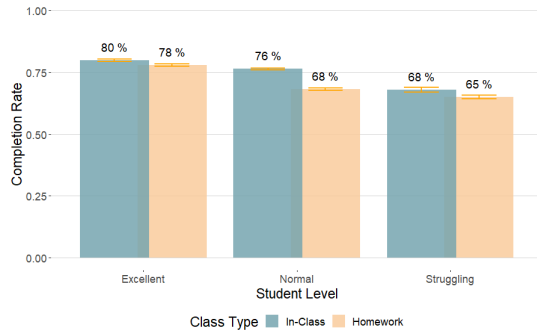


Figure 4: Completion Rate in English

The mastery rates by subject and student success level are presented in Figures 5 and 6. We conducted the Fisher’s

exact tests and it demonstrated that the excellent students had a better performance in terms of mastery rates compared to the normal students (Math, 68.6% vs. 52.1%, $p < 0.001$; English, 63.6% vs. 54.7%, $p < 0.001$). The mastery rates were much lower for the struggling students (Math, 36.9%, $p < 0.001$; English, 11.9%, $p < 0.001$).

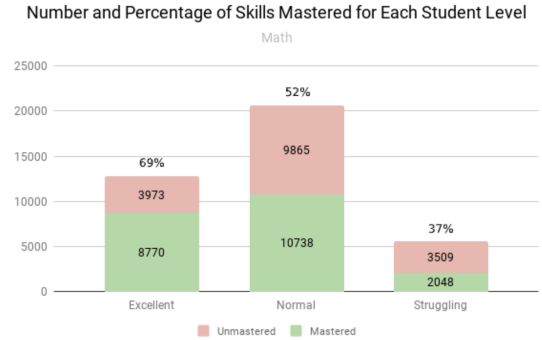


Figure 5: Mastery in Math subject

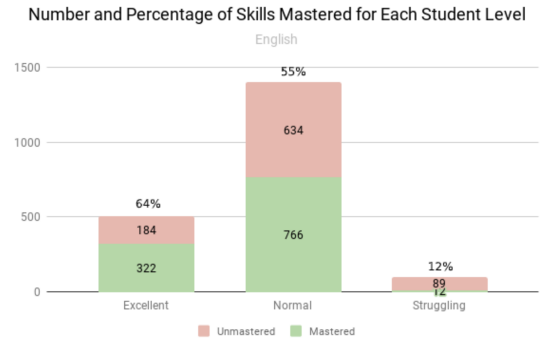


Figure 6: Mastery in English subject

Figures 7 and 8 show that the average mastery rate of RtL tasks was significantly higher than that of UtL tasks, $p < 0.001$ for each of the three student success levels in each subject, using Fisher’s exact test.

The logistic regression provided further evidence that ZPD status was associated with students’ learning outcome ($F(2, 38891) = 119.85, p < 0.001$ for Math and $F(2, 1996) = 30.74, p < 0.001$ for English), with adjustment for task type and student success levels. In particular, a RtL task was more likely to be mastered than a UtL task (Math, $OR = 1.710, p < 0.001$; English, $OR = 7.709, p < 0.001$), but was less likely to be mastered than an AM task ($OR = 0.241, p < 0.001$ for Math and $OR = 0.185, p < 0.001$ for English). Moreover, the logistic regression also suggested that students were more likely to master a math skill in class compared to homework ($t(38891) = 2.676, p = 0.007$), while the mastery rates of English skills were similar between the two settings ($t(1996) = 0.706, p = 0.480$).

Moreover, interaction terms were added to the logistic regression model in order to test the hypothesis that the relationship between ZPD status and learning outcome was

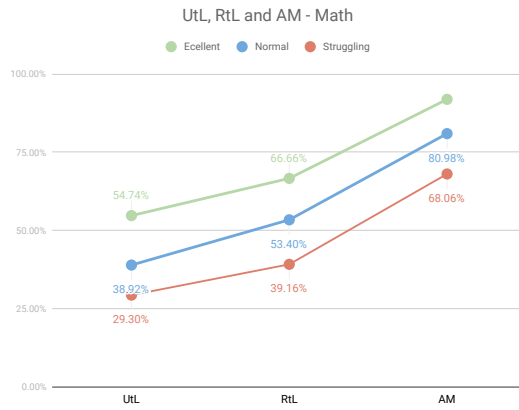


Figure 7: ZPD v.s Mastery in Math subject

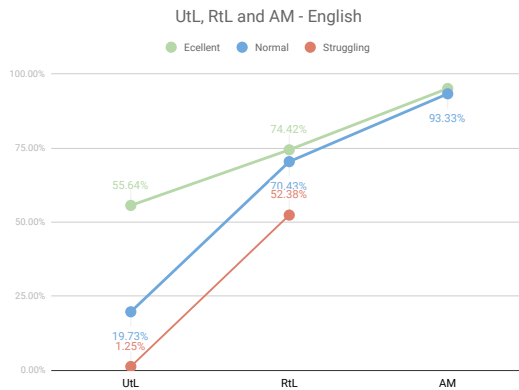


Figure 8: ZPD v.s Mastery in English subject

different for students with various success levels (i.e., excellent, normal, struggling). It turned out that, within the subject Math, the improvement on learning outcome associated with RtL status were comparable among the three student groups ($F(2, 30664) = 4.374$, $p = 0.126$), which was consistent with the observation that three lines corresponding to different success levels are almost parallel in Figure 7. Within the English subject, however, the analysis results indicated an interaction effect between RtL status and student success levels ($F(2, 1587) = 8.763$, $p < 0.001$): the excellent students tended to benefit less from being assigned a RtL task instead of a UtL task than either normal students ($p < 0.001$) or struggling students ($p = 0.002$). The conclusion with respect to AM tasks was less clear because there were fewer struggling students to begin with.

Lastly, we did not find statistical evidence for there being an interaction effect between ZPD status and whether the system was used in class or as homework ($t(38891) = -0.282$, $p = 0.778$ for Math and $t(1996) = 1.859$, $p = 0.063$ for English). This suggests that it is likely important to assign RtL content to students regardless of which setting the system is used in, although it may be warranted to continue investigating whether RtL content has more benefit for students studying English in class, based on the marginally significant

p value in that analysis.

6. DISCUSSION & CONCLUSION

In the light of these results, we can re-consider our original research questions. We hypothesized that, as in [10], students would master more tasks if presented with content thought to be in their ZPD (Ready-to-Learn content) than content outside of their current ZPD (Unready-to-Learn content). Our findings are compatible with this hypothesis, providing a replication of the earlier work in [10]. We also find that this pattern replicates across two domains, Math and English.

Our second hypothesis was that students would have higher mastery rates in class than when completing homework; this hypothesis was upheld for math subject but not upheld for English subject. Our finding is that students were slightly more likely to master a math skill in class than as a homework, while the mastery rates of English skills was comparable between the two contexts. This finding may suggest that the learning support within the platform was more effective than anticipated; alternatively, it may be that the instructors using the platform in their classes have not yet learned effective pedagogies for teaching students using this type of technology. Effective teaching in these contexts involves different pedagogies than are necessary within traditional classrooms [6], and there is increasing evidence that many teachers do not adopt these pedagogies until their second year of teaching with a new technology [5].

Our third research question asked whether generally more successful students would perform better than other students, even for content seemingly outside their zone of proximal development. In line with past work by Liu and Koedinger (2015) [2], it seemed that these more successful students were more able to succeed, even on this content that was anticipated to be highly difficult. However, they still performed more poorly on this content than on content thought to be in their ZPD.

Overall, these results suggest that assigning content with regards to a student's zone of proximal development can lead to a higher probability of the student mastering the content they are given. This result, a replication of [10], appears to hold in more than one learning domain. However, there are several important areas of future work before this finding can truly be held to be robust. First, this finding should be replicated in a broader range of contexts – other learning systems, other learning domains, and a wider range of learner populations and countries. Second, it is probably warranted to look at other definitions of the ZPD to refine this finding – is there an optimal degree of prior mastery for assignment of a student within the knowledge graph? Would alternate definitions of ZPD, such as seen in Murray and Arroyo's work (2002)[4], be equally or more effective? Does this type of finding also hold within systems where content is not consolidated into skills but is more factual in nature? By learning the answer to these questions, we can improve the effectiveness of adaptive learning systems more broadly, while helping to better operationalize and understand one of the classic theories in the history of thought on education.

7. REFERENCES

- [1] M. Feng and N. Heffernan. Informing teachers live about student learning: reporting in the assessment system. In *12th Annual Conference on Artificial Intelligence in Education 2005 Workshop on Usage Analysis in Learning Systems*, Amsterdam, 2005.
- [2] R. Liu and K. R. Koedinger. Variations in learning rate: Student classification based on systematic residual error patterns across practice opportunities. In *Proceedings of the International Conference on Educational Data Mining*, 2015.
- [3] J. A. Marsh, J. F. Pane, and L. S. Hamilton. Making sense of data-driven decision making in education. In *Evidence from recent RAND research*, Pittsburgh, PA, 2006. RAND Corporation.
- [4] T. Murray and I. Arroyo. Toward measuring and maintaining the zone of proximal development in adaptive instructional systems. In *Proceedings of the International Conference on Intelligent Tutoring Systems*, 2002.
- [5] J. F. Pane, B. A. Griffin, D. F. McCaffrey, and R. Karam. Effectiveness of cognitive tutor algebra i at scale. *Evaluation and Policy Analysis*, 36(2):127–144, 2014.
- [6] J. W. Schofield. *Computers and classroom culture*. Cambridge University Press, Cambridge, UK, 1995.
- [7] L. Staman, A. J. Visscher, and H. Luyten. The effects of professional development on the attitudes, knowledge and skills for data-driven decision making. *Studies in Educational Evaluation*, 42:79–90, 2014.
- [8] L. Vygotsky. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, UK, 1978.
- [9] A. F. Wise and Y. Jung. Teaching with analytics: Towards a situated model of instructional decision-making. *Journal of Learning Analytics*, 6(2):53–69, 2019.
- [10] X. Zou, W. Ma, Z. Ma, and R. S. Baker. Towards helping teachers select optimal content for students. In *Artificial Intelligence in Education*, 2019.

Mutual spontaneous help between students in an online learning environment and the role of the feeling of social belonging to a group

Dalila Bebbouchi

Laboratoire CIREL, équipe Trigone.

Université LILLE (France)

dalila.bebbouchi.etu@univ-lille.fr

Centre de Recherche sur l'Information

Scientifique et Technique

CERIST(Algérie)

dbebbouchi@cerist.dz

ABSTRACT

The communication presents a doctoral research currently underway in the context of adult e-education. It's interested in spontaneous mutual helping behaviors between learners engaged in fully online learning. It aims to identify the nature of this mutual help and to examine the influence of the feeling of social belonging on this mutual aid.

In order to give an overview of this thesis work, we were inspired by research carried out on mutual aid behaviors, prosocial behaviors and the feeling of social belonging. The basic psychological needs theory has also provided theoretical support for this research.

For the empirical study, we favored a comprehensive approach integrating a mixed methodology involving various sources of data collection and different methods of analyzing this data: correlation analysis on quantitative data from a survey questionnaire, lexicometric and thematic analysis on qualitative data from interviews and analysis of traces of mutual help on the platform's forums. The results of these analysis shed light on the perception of the feeling of social belonging and its role in helping behavior.

Keywords

Mutual help, feeling of belonging, distance learning.

1. INTRODUCTION

With the evolution of online communication tools, the learner engaged in an e-learning system has multiple possibilities to interact with his teachers, tutors or peers as part of his training. These interactions make it possible to create a socio-emotional climate favorable to transactions between learners (confrontation of their points of view, mutual adjustments,

negotiations) and to break the isolation [1]. The development of interactions between all actors of a device can be realized only if there is commitment between each other. In a context of adult e-education, engaging in a training project and persevering depends on several factors: individual psychological factors influenced by the social environment and the relationship with others, social experience, favorable or unfavorable dispositions training, personal and professional projects [2].]. It is in this particular context of adult education and elearning that this search is registered. It aims to describe spontaneous helping behaviors between learners i.e. helping behaviors initiated by the caregiver without having been invited [3] and to examine the feeling of belonging effect on these behaviors. In this perspective, our research aims to answer the following questions:

How do learners help each other? Are their mutual help behaviors linked to a sense of social belonging?

First, we are going to describe the theoretical framework. We will give the definition of our research dimensions, then we will detail the research methodology. Finally, we will present the results and the first conclusions.

2. THEORETICAL FRAMEWORK

2.1 Mutual help

Mutual help has been the subject of several researches in various fields and on different groups of people. Peer support experiences in primary and general secondary education, described either as "tutoring" or as "monitoring", cover identical practices. These are always mediation situations where a learner helps another learner in his academic, methodological learning and in the organization of his personal work [4]. In the context of e-learning, a helping relationship refers to tutoring where the help consists of psychological support which adopts empathy, active listening and non-judgment [5]. In their review of the literature on the learners e-learning experience, Dieumegard and Durand have shown that in several systems, learners tend to move away from institutionalized exchange spaces to help each other [6]. In addition, other research has also shown the emergence of mutual help networks at a given time in e-learning [7]. The analysis made on the exchanges between learners on the forums revealed, among other things, that these networks could consolidate the learning

Dalila Bebbouchi "Mutual spontaneous aid between students in distance learning and the role of the feeling of social belonging to a training group" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 754 - 757

process, recreate a space-time of meeting in order to reduce the "distance" and to overcome technical problems. Furthermore, in the context of work and organizations, helping behavior has been identified as an important form of organizational citizenship [8]. Paillé defines mutual help as a helping behavior which "consists for a person in providing assistance to a colleague to enable him to solve a problem or to avoid the emergence of difficulties encountered in the performance of his work" [9]. This helping includes four dimensions: altruism which is a voluntary actions that help another person with a work problem, peacemaking defines the actions that help to prevent, resolve or mitigate unconstructive interpersonal conflict; cheerleading which means the words and gestures of encouragement and reinforcement of coworkers' accomplishments and professional development and finally courtesy which involves helping others by taking steps to prevent the creation of problems for coworkers [8].

2.2 The feeling of social belonging

Several names for the concept of the feeling of social belonging are used in an undifferentiated way in the research work [10, 11, 1] such as "Affiliation", "relatedness", "Connectedness", "belongingness". This feeling cannot be formed individually [10], it can only exist if the individual is accepted and recognized by the other or more precisely by the members of the group with whom he wants to be and wishes to share his values [12].

The feeling of social belonging is expressed by two sub-dimensions [11]: the feeling of intimacy and proximity between two or more people expressed by the fact of being attached, united or friend with the other, the second sub-dimension refers to the feeling of acceptance which expresses the fact of being accepted, understood, valued, listened to or even in trust with the other. Acceptance by others leads to a variety of positive emotions (happiness, delight, well-being, calm) while being rejected, excluded or ignored leads to powerful negative feelings such as depression, grief, jealousy and loneliness. The emotions that people experience, which are both positive and negative, are linked to the feeling of belonging [13]. It should be noted that the feeling of social belonging is strongly linked to the need to belong. Indeed, the first is centered on others and the second is linked to the image given to others [14]. This need according to Deci and Ryan is part of the fundamental psychological needs inherent in human nature [15].

3. METHODOLOGY

This research is based on a comprehensive approach. It integrates several sources of data collection. Also, it takes into account different levels of analysis to obtain a richer understanding of the feeling of social belonging and its role in helping behavior. The chosen field of study concerns two promotions of the online training Master 2 "Multimedia Pedagogical Engineering" (IPM). This master is provided by the Department of Education Sciences and Adult Education (SEFA) of Lille University. The first promotion was at the beginning of the course and the second at the end of the course. The majority of learners are adults continuing their studies. Teaching is done through an e-learning platform ACCEL (Collaborative Learning and Online Community). ACCEL is an online learning platform for group animation based on organized asynchronous exchanges enriched with documents [16].

Data collection was carried out in three stages: a first stage aimed at examining the field through the administration of a survey questionnaire sent to 114 learners distributed as follows: 62 learners at the end of the course and 52 learners at the beginning of the course. The questionnaire was designed on the basis of psychometric scales validated theoretically and empirically: ESAS scale [11] which measures the feeling of social belonging and the mutual helping scale [8] which measures the helping behavior as defined by Podsakoff and al. [9].

The second data collection comes from semi directed interviews conducted by VoIP with a panel of 20 volunteer students: 12 students at the end of course and 08 learners at the beginning of course. The interview guide was designed considering the indicators of the feeling of social belonging and mutual help.

Finally, third collection of data was taken from the traces of mutual help between learners on the platform's forums, 1400 contributions were analyzed.

4. RESULTS AND ANALYSIS

The correlation analysis conducted on the quantitative results reveals that there are no significant links between mutual help and the feeling of social belonging at the beginning of the course (Table 1). There is a negative correlation between the indicators of mutual help and the feeling of intimacy. On the other hand, for respondents from the promotion at the end of the course, the analysis shows that there is a significant link between the feeling of intimacy (indicator of the feeling of social belonging) and peacemaking (indicator of mutual help) (table 2)

Table 1: Correlations analysis at the start of the course

	Acceptance	Intimacy
Altruism	0,121	-0,008
Courtesy	-0,075	-0,333
Peacemaking	0,034	-0,254
cheerleading	0,083	-0,257

* Significant correlation at level 0.05; ** significant correlation at level 0.01

Table 2: Correlations analysis at the end of the course

	Acceptance	Intimacy
Altruism	0,134	0,210
Courtesy	0,035	0,310
Peacemaking	0,066	0,508**
cheerleading	0,058	0,136

* Significant correlation at level 0.05; ** significant correlation at level 0.01

Learners who show altruistic behavior and donate their time, feel close to their peers and united with them. These results clearly show the development throughout the formation of socio-emotional relationships between learners.

The qualitative data, resulting from the transcription of 20 interviews, underwent a double analysis. The first is a textual statistical analysis and the second is a qualitative analysis by using the conceptualizing categories approach [17]. The categories were extracted from the literature and represent the different indicators of our two research dimensions.

A statistical textual analysis of the frequencies of the words used in the 19 transcribed interviews was conducted using the Iramuteq lexical analysis software. This free software studies groups of significant words and proposes groupings. Three major classes

emerge from the first analyses (Figure 1): A class characterized by the forms "diploma", "career", "professional" which reflects the professional trajectory of the learners as well as their motivation for training. A second class characterized by the forms "group", "phase", "integration" reflects the group organization aspect. The third class is rather related to the moods and feelings of the learners, there are forms related to the dimension of mutual help and the dimension of belonging. This analysis reveals that mutual help takes place in a small working groups.

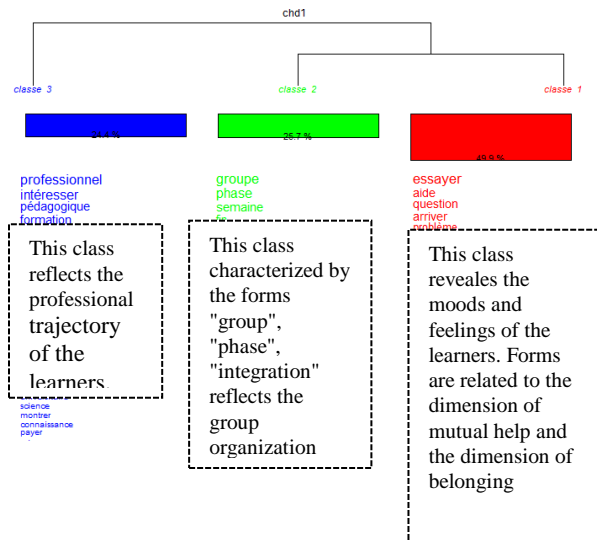


Figure 1. Descending hierarchical classification

A qualitative analysis highlights two group profiles, groups where, understanding and empathy prevail, and groups with difficulties related to both conflicts and agitations between the members of the working group. Given the results, the feeling of belonging is strong mainly in working groups where, understanding and empathy reign. We find in these groups, essentially, altruistic and cheerleading behaviors.

The table of data and variables (Figure 2) shows the results of qualitative analysis. It highlights two group profiles, groups where, understanding and empathy prevail, and groups which have experienced difficulties related either to conflicts and agitations between the members of the working group or abundance of one of the members group. Feeling of belonging is strong mainly in working groups where, understanding and empathy reign. We find in these groups, essentially, altruistic and cheerleading behaviors.

	Entraide				SAS			Groupe (entraide/compréhension)	Groupe (conflit/agitation)
	Altruiste	Conciliation	Courtoisie	Réconfort	Acceptation	Intimité			
E1	x			x	x			x	
E2				x	x				x
E3	x		x		x		x		
E4	x			x	x		x	x	
E5									x
E6	x		x		x		x		
E7		x		x	x	x	x		
E8	x			x	x	x	x		
E9	x			x	x		x	x	
E10		x			x			x	
E11	x	x		x	x		x		
E13	x	x		x	x		x		
E14	x	x		x				x	
E15	x			x	x			x	
E16	x	x		x	x	x	x		
E17	x			x	x				x
E18	x			x	x	x	x		
E19	x				x			x	
E20	x		x		x	x	x		

Figure 2. Data table and variables

Traces analysis of mutual help on the platform's forums is still in progress. 1400 contributions on the thematic forums of the promotion at the beginning of the course were browsed in order to identify those, which describe mutual helping behavior between learners. The first results reveal a weak tendency of learners to provide help to their peers. Requests for help are rather directed towards tutors. The rare helps provided spontaneously by peers categorized as altruistic behaviors mainly concern the organization of the platform such as access to a space, production depot, access to documents. It seems that requests for help are made through non-formal networks [7] or via communication tools other than those present on the platform.

5. CONCLUSION

The mutual help behaviors as we have apprehended them are found in the small working groups, the latter are divided into two groups. Groups in which there is a stable climate of understanding and empathy and groups that have broken up because of the abundance of one of their members, or in which the work has been carried out in anguish and agitation. In the first groups, there is a strong feeling of social belonging (acceptance and intimacy). Learners feel understood and supported and help each other all the time. These behaviors are present at the end of the training course. It seems that the fact of having shared several activities collectively during the whole training favored the development of interpersonal, intimate and regular relationships. However, the feeling of social belonging is low in the groups that have experienced unrest situations or that have broken up due of their member's abandonment. In view of the results, it seems that a third dimension on group dynamics is at the origin of the feeling of social belonging development and the appearance or not of mutual help behaviors.

6. REFERENCES

- [1] Jézégou, A. (2010). Créer de la présence à distance en e-learning. Cadre théorique, définition, et dimensions clés. *Distances et savoirs*, 8 (2), 257-274. <http://www.cairn.info/revue-distances-et-savoirs-2010-2-page-257.htm>
- [2] Gausse, M. (2011). Se former tout au long de sa vie d'adulte. Dossier d'actualité Veille et Analyse, 61. <http://veille-et-analyses.ens-lyon.fr/DA/detailsDossier.php?parent=accueil&dossier=61&lang=fr>
- [3] Ros, J., Grossen, M. (2016). L'entraide en institution pour personnes en situation de handicap mental : d'une recherche-action à un modèle d'analyse. *Les Cahiers Internationaux de Psychologie Sociale*, 110, 137-158.
- [4] Bédouret, T. (2003). Autour des mots "Tutorat", "Monitorat" en éducation. *Recherche et formation*, 43. <http://ife.ens-lyon.fr/publications/editionelectronique/recherche-et-formation/RR043-08.pdf>
- [5] Rodet, J. (2011). Formes et modalités de l'aide apportée par le tuteur. C. Depover, B. De Lièvre, D., J. J. Quintin, et A. Jaillet (Eds.). *Le tutorat en formation à distance* (p. 159-170). Bruxelles, De Boeck Supérieur.
- [6] Dieumegard, G., Durand, M. (2005). L'expérience des apprenants en e-formation : revue de littérature. *Savoirs*, 1(7), 93-109. <http://www.cairn.info/revue-savoirs-2005-1-page-93.htm>

- [7] Foucault, B., Metzger, J. M., Pignorel, E., Vaylet, A. (2002). Les réseaux d'entraide entre apprenants dans la e-formation : nécessité et efficacité ? Education permanente, 152. <https://halshs.archives-ouvertes.fr/edutice-00000309/document>
- [8] Podsakoff, P. M., & MacKenzie, S. B. (1994). Organizational Citizenship Behaviors and Sales Unit Effectiveness. *Journal of Marketing Research*, 31, 351-363.
- [9] Paillé, P. (2007). La citoyenneté dans les organisations. Validation française des échelles de mesure de Podsakoff et MacKenzie (1994). *Les Cahiers Internationaux de Psychologie Sociale*, 74 (2), 59-66. <http://www.cairn.info/revue-les-cahiers-internationaux-de-psychologie-sociale-2007-2-page-59.htm>
- [10] Goodenow, C. (1993). The Psychological Sense of School Membership among Adolescents : Scale Development and Educational Correlates. *Psychology in the Schools*, 30(1), 79-90.
- [11] Richer, S., Vallerand, R. (1998). Construction et validation de l'Echelle du sentiment d'appartenance sociale (ESAS). *Revue Européenne de Psychologie Appliquée*, 48(2), 129 – 137.
- [12] Francard, M., Blanchet, P. (2003). Sentiment d'appartenance. In G.Jucquois & G.Ferréol(Eds.), *Dictionnaire d'interculturalité* (p. 18-25).Paris : Armand Colin.
- [13] Baumeister, R. F., Leary, M. R. (1995). The Need to Belong: Desire for Interpersonal Attachments as a Fundamental Human Motivation. *Psychological Bulletin*, 117(3).pp. 497-529.
- [14] Sanquirgo, N., Oberlé, D. et Chekroun, P. (2012). L'échelle de besoin d'appartenance : validation française et rôle dans les réactions à la déviance. *L'Année psychologique*, 112(1), 85-113. doi:10.4074/S0003503312001042.
- [15] Deci, E. L., Ryan, M. L. (1985). Intrinsic motivation and self-determination in human behavior. Springer Link. <http://link.springer.com/book/10.1007%2F978-1-4899-2271-7>
- [16] Delache, D., D'Halluin, C., Fichez, E., Hoogstoel, F., Leclercq, G., et al.(2006). *Environnements numériques et pratiques collaboratives d'apprentissage* [Compte-Rendu de fin de recherche de l'opération PCDAI]. <http://www.univ-lille3.fr/fr/recherche/equipes-recherche/geriico>
- [17] Paillé, P., Mucchielli, A. (2012). *L'analyse qualitative en sciences humaines et sociales*. Paris, Armand Colin.

Structural Explanation of Automated Essay Scoring

Afrizal Doewes, Mykola Pechenizkiy
Eindhoven University of Technology
{a.doewes, m.pechenizkiy}@tue.nl

ABSTRACT

Scoring an essay is an exhausting and time-consuming task for teachers. Automated Essay Scoring (AES) facilitates the scoring process to be faster and more consistent. Nevertheless, AES system lacks transparency about the reasoning behind the score given to the students. This research aims to find a suitable framework for providing an informative score explanation. In our experiment, we develop a regression model using Gradient Boosting, then analyze the overall features contribution and local interpretation of the score prediction. We construct the feedback summary by decomposing the feature contributions and categorizing similar features into a structural explanation. The results indicate that structural explanation can help researchers to recognize and improve the performance of the system when dealing with problems such as gibberish, autocorrect, and spelling errors. The feedback can also highlight the strength and weakness of a student's answer.

Keywords

Automated Essay Scoring, Structural Explanation, Feature Contribution

1. INTRODUCTION

There is a growing interest to use computer software as tools to facilitate the evaluation of student essays. Theoretically, Automated Essay Scoring (AES) system works faster, reduces costs in terms of evaluator's time, and eliminate concerns about rater consistency. However, AES system lacks transparency about the reasoning behind the score prediction. It is highly needed to build trust in machine learning models trained for classroom contexts [1]. Furthermore, AES system must provide good quality and useful feedback to its users, which can be inspired by the field of Learning Analytics. Researchers from the University of Technology Sydney, Australia, are designing personalized and automated feedback to develop students' research writing skills [2]. They develop a system called AcaWriter for providing formative, actionable feedback on HDR (Higher Degree Research) student writing. The system implements a genre-based approach and the CARS model [3], which describes the rhetorical and linguistic patterns that authors make in their research article introduction. The students stated that AcaWriter helped them think about the structure of their article introduction and focus on the rhetorical moves in their writing. They also found that immediate feedback and text highlighting in the system useful. Pigaiwang [4] is another system providing feedback which is used in more than 1000 schools

in China, including some top universities, such as Tsinghua University, Nanjing University, Fudan University, and so on. Pigaiwang has made an essential contribution to English writing education at university. Pigaiwang provides students with opportunities to revise their writing and continues giving feedback, which improves their writing ability. Revision Assistant is another work which is a tool for providing sentence-level and rubric specific feedback to students [5].

The system feedbacks from previously mentioned studies are mostly provided in the revising phase. Students are expected to revise their work in order to get a better score. In this research, we focus on the final score feedback, which explains to students why the system gives them the generated score. Students are not able to revise their works, but the students can still take advantage of such feedback to perform better in their future exam.

The main contribution of this paper is to enable an AES explanation framework reproducible for researchers to develop their AES system. Unlike the proprietary systems, we develop our system in a transparent way by using open-sourced libraries. We use open and free libraries for the feature extraction, machine learning model training, and the model interpretation. This paper begins with the motivation for finding a suitable framework for score explanation. Then, we present the proposed framework and the experiment settings for generating the score feedback from feature contributions. Afterwards, we discuss the experiment results, system evaluation and improvement. Finally, we conclude our research and plan our future work.

2. PROPOSED FRAMEWORK

Figure 1 describes how the system works. By the time the student submits his/her answer, the raw text answer will be extracted into a feature vector. The regression model will then predict a score for this specific feature vector. The score prediction should be accompanied by the reasoning behind the score in the form of feedbacks. The feedbacks should highlight the strengths and weaknesses of the answer. The strengths are summarized from the feature categories with positive contribution towards the score, and the weaknesses are summarized from the ones with negative contribution.

Feedback in AES system provides transparency about the grading process. This can ensure fairness for all students and make sure that each students' essay is evaluated by the same standard. Students can also identify their strength and weakness, which is beneficial for their future exam. Teachers can take advantage of the feedback feature in AES to assess the performance of the system, and to check whether specific learning objectives have been fulfilled. Score explanation also enables researchers to evaluate and to improve the performance of their AES system by analyzing the model interpretation behind the score prediction.

Afrizal Doewes and Mykola Pechenizkiy "Structural Explanation of Automated Essay Scoring" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 758 - 761

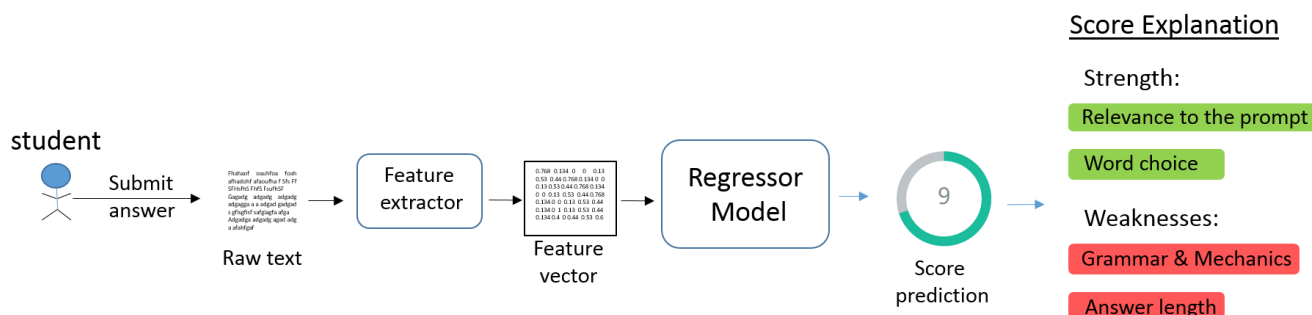


Figure 1 Score Explanation for AES Framework

3. SCORE ANALYSIS FROM FEATURE INTERPRETATION

We develop our Automated Essay Scoring model using Gradient Boosting algorithms. Ensemble model such as Gradient Boosting (GB) is especially hard to interpret because of the complexity. The trade-off between model performance and model interpretability is known among researchers. Generally, a more complex model outperforms a simple linear model. Therefore, we choose to understand the model decision using several interpretation techniques rather than sacrifice the system performance.

3.1 Overall Feature Interpretation

Using XGBoost library, we can train the model and also extract the importance of the features from our model. Identifying the essential features can help us in understanding the behavior of the model in general.

3.2 Score Analysis from Local Interpretation

Local interpretation means that we are interested in understanding which variable, or combination of variables, determines the specific prediction. We use shap values to help in determining the most predictive variables in a single prediction. In AES, the system output is a real number. Each variable contribution will either increase or decrease the output value.

4. EXPERIMENTS

4.1 DATASET

We use the Automated Student Assessment Prize (ASAP) dataset¹, hosted by the Kaggle platform, as our experiment data. In this research, we use specifically dataset #6 from ASAP. The dataset comprises 1800 essays, which then split into the training set and testing set in 80:20 ratio. The score range in this dataset is 0 – 4.

4.2 FEATURES EXTRACTION

The essay features are extracted using EASE (Enhanced AI Scoring Engine) library², written by one of the winners in ASAP Kaggle competition. This features set have been proven to be robust [6]. EASE generates 414-length features. We added one more feature (spelling error) later at the evaluation phase, so that we have 415 features in total.

4.3 MODEL TRAINING

We train the regression models using Gradient Boosting algorithms. We use Quadratic Weighted Kappa (QWK) score as the

evaluation metric. QWK measures the agreement between system predicted scores and human-annotated scores. The mean QWK score for our Gradient Boosting (GB) model using 5-fold cross validation is 0.7667.

5. RESULTS

5.1 Overall Features Interpretation

XGBoost Python package includes the plotting function to reveal the importance of each feature from the model. We show 15 features with the highest importance. Answer length appears to be the most important feature in predicting the essay score. Average word length, prompt overlap ratio, and good n-gram ratio are also among the most important features. Meanwhile, some of the other features are not interpretable because they are merely the bag-of-words representation of the answer. We did not eliminate the bag-of-words features because the model performance, indicated by mean QWK score, is slightly lower without their presence.

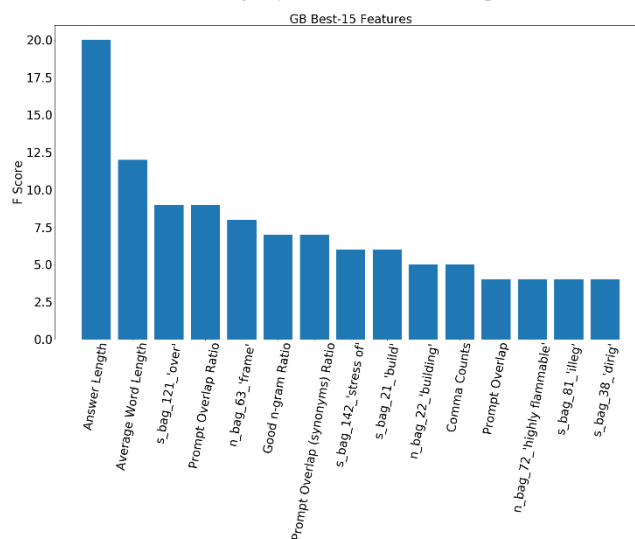


Figure 2 The 15-most important features from Gradient Boosting

5.2 Local Interpretation

Local interpretation deals with a single instance prediction, it helps us to analyze the reasoning behind the model prediction. Figure 3 shows each feature's contribution to obtain the score prediction from an essay in the test set. We examined the prediction of essay

¹ <https://www.kaggle.com/c/asap-aes>

² <https://github.com/edx/ease>

sample from the ASAP dataset #6 with essay ID: 15360, taken from the testing set. This answer has a score of 3 out of 4, which is the correct prediction. We can observe that the most influential contributor in predicting the score is the answer length, which has the largest impact on increasing the score. It seems that the student wrote his/her answer above the average length of the other answers. There is a tendency that a longer answer is generally awarded a higher score. Although it remains unclear whether longer essay also provides better ideas and arguments.

Prompt overlap is the second interpretable feature that also improves the score. Prompt overlap means the number of same tokens that are found between the answer and the prompt. Too high overlap score might indicate that the student is not creative or original enough in writing his/her own ideas and words as the answer. However, too low overlap score is also a warning that the answer might be out of topic.

Meanwhile, the average word length affects negatively to the score. Average word length feature can provide an insight that longer word could mean a more sophisticated word choice and help the students to achieve a better score.

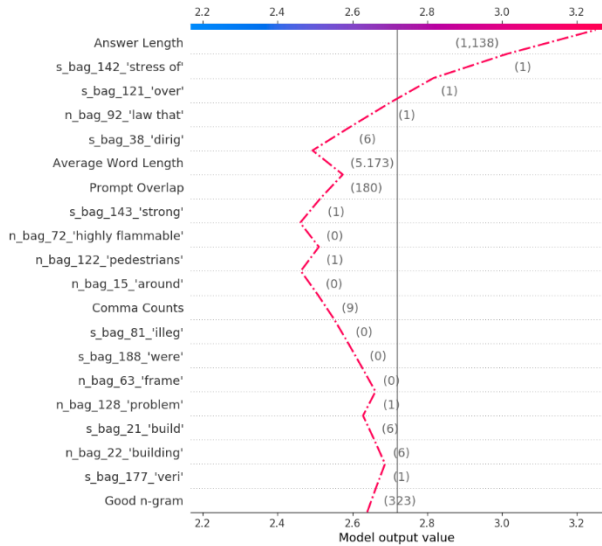


Figure 3 GB Feature Contribution for essay ID: 15360

5.3 Structuring the Feedback

We believe that categorizing the feedback in a more structural way is better and can provide a higher level of feedback to the users. Therefore, we propose our structural explanation of score prediction by AES system.

Our framework explains the score prediction in five categories, as we can see in Table 1. The features in the second column are from EASE library, plus one spelling error feature, which we added later in the evaluation and improvement part. Each feature has a different contribution value; it can be either positive or negative. The feedback summary in the first column categorizes similar features and gets its value by summing the contribution values of those features. The summation results with negative values belong to negative feedback, and the ones with positive values belong to positive feedback.

Our first category deals with answer length, and it is the sum of the contribution values of two features; answer length (number of total characters in the answer) and word counts. Relevance factor combines four features from EASE which are related to the degree

of overlap between the prompt and the answer, including the synonyms. Grammar measures the number of good n-gram and its ratio in the essay. The essay is extracted into its POS-tags and we compare them with a list of valid POS-tag combinations in English. The usage of punctuation in the answer, combined with how many spelling errors found, defines the mechanics feedback. Under the assumption that a longer word means a more difficult or sophisticated word, we put the contribution of feature average word length in its own category, namely Difficult Word Usage.

Table 1 Feedback Categories for Score Explanation

Feedback Summary	Contributing Features
Answer Length	- Answer Length - Word Counts
Relevance	- Prompt overlap - Prompt overlap ratio - Prompt overlap (synonyms) - Prompt overlap (synonyms) ratio
Grammar	- Good n-gram - Good n-gram ratio
Mechanics	- Comma Counts - Apostrophe Counts - Other punctuation counts - Spelling errors
Difficult Word Usage	- Average word length

Categories with positive contribution are shown in green. On the other hand, categories which are proven to be negatively affecting the score are displayed in red. We exclude the bag-of-words features from our feedback summary because they are less interpretable. Feedback for essay ID: 15360 is shown in Figure 4.

5.4 Evaluating and Improving the System

It is important to note that all of our feedbacks are based on the general assumption about the text features, and what we can infer from them. In the dataset (ASAP Dataset#6), the final scores are not accompanied by rubric scores or scoring criteria. Thus, we cannot understand the actual reasoning behind the scoring process by the persons who annotate the data. Therefore, we come with our proposed solution to provide score explanation from text feature extraction and see their contribution from the model interpretation. Based on that condition, we can only test our system using some extreme essay samples. The reason is that we are looking for examples that we are confident about the score that should be given.

We can observe three examples of inaccurate predictions or feedbacks from the system in Table 2. The first example (Answer ID: 1) test the system's ability to handle gibberish. We want to avoid users from tricking the system using invalid answers, and undeservedly get a score other than zero. However, the system incorrectly awards the first answer with a score of one. Using our framework, it is possible to analyze the cause of a wrong prediction. The feedback summary in Figure 5 (left) shows that this answer has positive feedback from difficult word usage category. The reason is that the gibberish contains many words with high average word length, which indicates the usage of difficult words from the users. And the usage of more sophisticated words tends to improve the user's score.

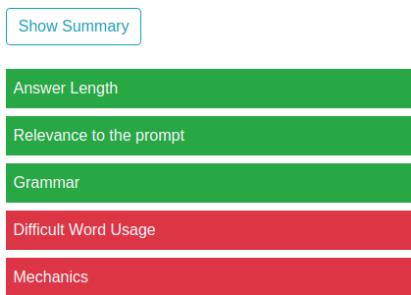


Figure 4 System Feedback for Essay ID: 15360

To improve the system, we modified one of our feature in the feature extraction phase. The model will only count the average word length for valid English words. We use Python spellchecking library PyEnchant³ to validate whether each word belongs to English vocabulary. Modifying this feature is able to correct the system prediction. The first answer gets the score of zero, and the system displayed the correct feedback summary, as shown in Figure 5 (right).

Table 2 Evaluating Wrong Predictions

Answer ID	Problem	Actual Output	Expected Output
1	Long gibberish	1	0
2	Long gibberish with inaccurate spell correction	1	0
3	Perfect score (4 out of 4) for an essay that have too many spelling errors	4	3

The second essay (Answer ID: 2) suggests that gibberish possess another form of risk. It seems that the autocorrect feature inside EASE library (Aspell spell checker) may transform the gibberish into a valid word. In the second essay, the sequence of characters such as “sigsigisghsi” is transformed into “zigzags”, “emoybgat” into “embark”, and “adjghadoigda” into “adjudicate”. These valid words, although not meant by the user, increase the average word length value which is correlated to difficult word usage category. Based on this problem, we decided not to implement spell correction while counting the average word length feature. Whereas, spell correction is still applied for the other features. Finally, the system is able to provide the expected prediction for the second answer, which is also zero.

The third answer is actually from the testing set (Essay ID: 15073), and it has the perfect score of 4 out of 4. However, we edited this answer so that it has many spelling errors (15 words). We cannot clarify whether spelling errors is influential in the score according to the human expert who annotated this data. However, we assume that any answer which has that many spelling errors should not be awarded a perfect score. For this reason, in addition to EASE features, we include one more feature, namely spelling errors. It counts the number of spelling errors that appear in the submitted answer.

We rebuilt the Gradient Boosting model with 415 features (414 features from EASE + 1 spelling error feature). The new mean QWK score is 0.7623. Interestingly, the spelling error feature also appear in the top-15 features with the highest importance for the model. Finally, our new model predicts the third answer (Answer ID: 3) with the score 3 out of 4. Moreover, the spelling error feature has the highest negative contribution to the final score for this answer.

6. CONCLUSION AND FUTURE WORK

The purpose of this research is to develop an Automated Essay Scoring (AES) system that can be used in practice. We focus on the score explanation aspect of AES. We demonstrated that our structural explanation framework can be beneficial for researchers to evaluate and to improve the performance of an AES system. Our experimental study shows that by analyzing the system explanation feedback, we can detect faulty behavior of the system prediction such as when dealing with gibberish, autocorrect, and spelling errors problems. Nevertheless, since little is known about the effectiveness of the model and the features for application in different domains, we plan to investigate the suitable design for an adaptable domain setting in the future work. Our current approach still lacks the pedagogical aspects of essay scoring. This is our other future work direction that we expect to improve the system in general and presentation of the focused feedback in particular, thus being more helpful for teachers and students.

7. REFERENCES

- [1] P. West-Smith, S. Butler and E. Mayfield, "Trustworthy Automated Essay Scoring without Explicit Construct Validity," in *AAAI Spring Symposia*, 2018.
- [2] S. Abel, K. Kitto, S. Knight and S. B. Shum, "Designing personalised, automated feedback to develop students' research writing skills," in *ASCILITE 2018 - Open Oceans: Learning without borders*, Geelong, 2018.
- [3] J. Swales, *Genre Analysis: English in Academic and Research Settings*, Cambridge University Press, 1990.
- [4] Y. Liu, "A Research on the Application of Automatic Essay Scoring System to University's English Writing Education in the Era of Big Data: Taking Pigaiwang as an Example," *Studies in Literature and Language*, vol. 10, pp. 84-87, 6 2015.
- [5] B. Woods, D. Adamson, S. Miel and E. Mayfield, "Formative essay feedback using predictive scoring models," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.
- [6] P. Phandi, K. M. A. Chai and H. T. Ng, "Flexible domain adaptation for automated essay scoring using correlated linear regression," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

³ <https://pyenchant.github.io/pyenchant/>

Natural Language Processing for Open Ended Questions in Mathematics within Intelligent Tutoring Systems

John A. Erickson
Worcester Polytechnic Institute
100 Institute Road
Worcester MA, 01609
jaerickson@wpi.edu

ABSTRACT

Intelligent tutoring systems continue to enable teachers insight into their students in an immediate fashion. With deep fine-grained data provided to the teachers, they can gain a deeper understanding of the student's learning. While multiple systems exist, most are limited to specific, close-ended questions; these include questions with a set of known acceptable answers, such as solving for 'x' in an equation (i.e. in ' $x+4=6$ ', the clear answer would be 2). Questions of this variety are implemented within these systems and allow for timely feedback to the students. A system can easily decipher certain values to be incorrect answers and help can be offered to the student. While close-ended problems provide a wide range of insights into the student's process, they are often unable to gain the deeper discernment of the student's understanding. Open response questions elicit a greater scope of the student's understanding. However, very few intelligent tutoring systems provide support to teachers and students for these types of questions. Within the few that can, they are not able to offer automation for the process. One of the greater appeals of computer-based systems is that they provide teachers automated grading and give students immediate feedback. It is therefore my goal to further the study and development of automated assessment and feedback tools to support open-ended problems within computer-based systems. Toward this goal, my focus of research is on the development and deployment of automatic grading models, exploration of fairness within such models, and expansion of existing systems to leverage this research.

Keywords

Natural language processing; machine learning; word-embeddings; intelligent tutoring systems; automated grading; automated feedback

1. INTRODUCTION

Intelligent tutoring systems (ITS) have been around for some time, and their benefits have been discussed and noted in

studies such as [13][17]. These benefits, however, have been limited to close-ended problem types. As such, problems with close-ended answers are at the core of most ITS; including ASSISTments [5], McGraw Hill's ALEKSTM and Carnegie Learning's Cognitive TutorTM. This limitation comes from the overall goal of ITS; to provide automated feedback to students and timely reports to teachers about their students. Questions with close-ended answers allow these systems to achieve this goal. For instance, its very simple to set up a system to understand the correct answers when 1/2 or .5 are the only acceptable student answer. Studies such as [14] have discussed why multiple choice questions (close-ended questions) are so appealing: they're easy, accurate and timely to grade. While it is evident that the teachers gains a substantial understanding of the students comprehension from these questions, there is more to student's process of thinking. If the student selects A, the teacher can assert the student's rationale; however, this is a summation from other students selecting the same answer. Open responses questions provide students the opportunity to explain their own personal rationale; giving teachers an even more in depth understanding of the student's process of thinking. Studies such as [6] called attention to the fact that there are vast advantages to a greater spectrum of questions types; when focusing on evaluations with a single question type, it's insufficient in testing the students actual understanding and rationale/critical thinking. By providing support for open response questions, teachers are able to discern, in greater detail, what point the student became confused or if they ever understood. This is also supported by [7] which discussed the wider range of cognition required with open response questions as compared to close-ended multiple choice questions. However, as mentioned earlier, few intelligent tutoring systems support this type of question.

While not the only system to support open response problems, ASSISTments, the system through which much of my prior research has been conducted, is developing tools to improve the support of these problems for teachers. The capability to automatically grade student answers or provide immediate feedback to students is still lacking in comparison to what is possible for close-ended problems. For open response questions, natural language processing (NLP) must be utilized to provide such tool. Additionally, the infrastructure needs to be in place to support these machine learning algorithms for real-time use within classrooms.

John Erickson "Natural Language Processing for Open Ended Questions in Mathematics within Intelligent Tutoring Systems" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 762 - 765

Table 1: Rasch Model Performance from Erickson et al., 2020

Model	AUC	RMSE	Kappa
Rasch Model with teacher component	0.696	1.09	0.162
Rasch Model without covariates	0.827	0.709	0.370
Rasch Model with number words covariates	0.829	0.696	0.382
Rasch Model number words and Random Forest covariates	0.850	0.615	0.430
Rasch Model number words and XGBoost covariates	0.832	0.679	0.390
Rasch Model number words and LSTM covariates	0.841	0.637	0.415

In this paper I will be discussing my previous work, which has attempted to develop machine learned models for automatically grading student open response questions within ASSISTments, in addition to current and proposed future projects pertaining to the further study and development of tools to support these problems in classroom settings. Among this proposed future research, I will describe my intention to study similarity measures to allow students to see similar open ended response rationales to theirs; in this regard, I have drawn inspiration from an existing system, known as myDALITE [2], and propose an extension of this idea utilizing open-ended response problems.

2. PREVIOUS CONTRIBUTIONS

It's clear there is an advantage to developing a tool which can assist in automating open responses in mathematics within intelligent tutoring systems (ITS). If we can bridge the gap between the ITS capabilities within close-ended problems and open response problems, we can further empower teachers with a deeper knowledge of their student's logic. With this, I focused on starting from the ground up. Exploring our ability, within ASSISTments, to automatically grade open response student answers within open response questions in a mathematical domain.

2.1 Automated Grading

While others have utilized a multitude of NLP approaches to interpret and grade open response questions, [16] [15] [12] [18], most have been working with non-mathematical content. Much of the NLP research has consisted of essays and sentences with a standard corpus. This is why so many approaches looked to utilizing deep learning approaches, such as word embeddings Word2Vec [8] and GloVe [10] to gain a vector relational understanding of words. For my research [3], I set out to automatically grade open response questions within the mathematical domain. Contrary to previous research, the corpus within this study was unique in the sense that student answers would be a diverse assortment of words and mathematical functions. Not only was the corpus diverse in words and functions, but the answers were diverse in length. Some student answers consisted of one or two words, while others responded with multiple sentences.

Within this research, the route was taken to approach the NLP task with a wide variety of approaches and methods. With models developed from traditional NLP approaches such as a term-frequency inverse document frequency, *tf-idf* (bag of words model which counts the number of occurrences of the word and re weights the word), to deep learning approaches with word embeddings, a wide spectrum of approaches were attempted.

Overall, 6 different models were developed to predict the student's grade on an open response mathematics question. In Table 1, the baseline model was a Rasch model which didn't take into account any NLP developed models. From there, we supplemented the Rasch model with a *teacher component* and *number of words* covariate. Each of those performed worse than either the *tf-idf*, or the word embedding approaches. By augmenting the models with NLP approaches, the Rasch model was able to improve and provide a stronger performing model with our data (c.f. [3] for further detail pertaining to this study and analyses).

3. CURRENT WORK

While the top performing model in my previous study showed promise with an AUC of .850, RMSE of 0.615, and Kappa 0.430, beating the baseline and all other models, it was decided to ensemble the 3 top performing models. The ensemble, along with the individual previous 3 top models, are now currently being used within a randomized control trial and integrated within ASSISTments. What has become more and more evident is that when utilizing pre-trained word embeddings, there needs to be close consideration of model fairness. As studies such as [1] noted, there can be underlying biases within word embedding models.

3.1 Assessing Fairness

Since multiple of the models within the automated grading study utilize pre-trained word embeddings, my research has progressed towards exploring potential bias within our models. Its imperative that models being implemented within an ITS, or any study, should minimize bias; especially as it pertains to grading. The grades should be based solely on the content, nothing else. As stated earlier, [1] notes that it doesn't matter which embedding approach you use (or pre-trained embeddings in our case), biases, such as gender bias, can sneak in. As the paper references, embeddings can teach models that woman is to homemaker as male is to computer programmer. This is something we explicitly want to avoid in any predictive models within an ITS.

Currently, work is being done to identify potential bias within models from my previous automated grading study. What is imperative is to be able to clearly identify the bias, if there is evidence of bias, and if its coming from pre-trained word embedding (when we account for the different word usages of males and females) or the models the grade predictions are trained with. By developing steps to directly compare models, and word representations, to predict grades given women responses/male responses, we can hopefully identify whether bias is present. We are building our approach from prior works (c.f. [4][9]), and if we can clearly identify which

models have the least amount of bias, then we push those models to production. Additionally, we will be exploring how to handle the bias, if needed, within the suspected models.

3.2 Randomized Control Trial

Currently, my research also is simultaneously being applied to a randomized control trial. This is a study in which the automated grading models are being used to provide student's with their potential grade before they submit an answer. So, once the student's have submitted a answer to the open response mathematics question, one of the conditions will take the strongest performing grade prediction model for the problem and suggest a grade. This grade is then presented to the student and the student will be presented with the option to edit their answer. This poses many interesting questions such as: will the student's edit their answers? If they do, by how much have the answers changed and how much has their grade changed. This is ongoing research and I will continue to develop new models and take into consideration the bias study previously discussed here.

3.3 Comment Suggestions

As discussed previously, one of the main attractions to ITS is the automation. While I have presented multiple models that predict the students grade with reasonable accuracy, within our data, it is clear there is another step. Providing automated feedback is the next optimal tool for teachers and students. Currently, work has been done developing an approach which suggest responses by utilizing similarity calculations. Recently, our team collected data where teachers graded a set of student open response answers. This allowed us to have multiple teachers grade the same student answer, as well. Within this, teachers would grade and create a category which they would place the student answer in. This was performed across multiple problems.

With this, there is now a more robust dataset of answers and associated teacher responses. By utilizing similarity calculations, ranging from Levenshtein distances to SBERT [11], when a student submits an answer to a problems (one which we have previous data on) the most similar student answer on file is calculated and we then can suggest those associated teacher responses with that most similar answer.

Additionally, its being explored how these methods could be validated. For instance, aside from manually looking at the suggested responses, how could there be an offline evaluation of these methods (that does not require teachers to select from the undoubtedly poor suggestions produced by early iterations of such a tool). For each problem, the 3 most similar answer for each individual answer (which has been graded and categorized by our teachers) are selected using both SBERT and Levenshtein distances. From there, it is calculated how many of the teacher categories are the same for the similar answer and the original answer. The method with the most agreement, for each problem, is selected to use for future student answers for said problem.

4. FUTURE WORK

With accurate grade prediction models, a potential method to identify bias, and an approach to selecting similar student

answers, I have a set of approaches which lends itself to the next step I wish to take. I am looking to explore whether we can expand upon just suggesting the student to go back and edit (the randomized controlled trial); can we use NLP to take the students answer, discover which are the most similar, find those similar answers and share their rationale with the student. Then allowing the students the opportunity to go back and either chose their submission or re-write their answers to reflect what they have learned from other similar (or possibly dissimilar) answer rationale. This requires a similarity calculation, a grade predictions (to see if the student's answers and most similar answer would retain the same or different grade) and then a way to show are calculations are accurate. Then once the student's answer has a top 3 similar student answers, the rationale (not answers) are shared. As identified earlier, this practice is in-part analogous to how an existing system, myDALITE [2] functions. It is for this reason that these same methods might be suited to expand upon this idea to provide teachers with new tools that can be used in the classroom.

In this system, students are presented with a multiple choice question and asked to provide an explanation, or rationale, for their work. Students are then presented with other rationales and asked if they would like to keep their answer or if a rationale for a different response has convinced them to change their answer. I wish to explore if this approach could be performed with open response questions. Instead of an initial multiple choice question the student writes a answer and rationale to an open response question and then similar responses are presented, giving the student the option to either change their response or continue. This would require multiple of my previous and current research to prepare such a approach.

This would be a fascinating exploration into how confident a student is in their response. If after seeing others rational, does that convince students to re-evaluate or edit their answers? We may be able to explore what types of answers are confident answers and how much they differ from less confident answers. Additionally, I would like to continue to use NLP to help identify gaming behavior with this type of system; it would be important to identify students answering with "I don't know" types of responses and avoid them simply being presented with other rationales. There are also questions into whether seeing other's rationale could hurt the students learning and cause more confusion. This is an aspect of the study which would need to be expanded upon.

Overall, there have been direct effects of my research, including the implementation of the automatic grader in ASSISTments using the models built in my previous research. Additionally, the current RCT provides an opportunity to see how these predicted grades could impact a student's answer if they were exposed to the grade. Lastly, there is potential for my work calculating similarities between student answers to impact how ASSISTments suggest responses for teachers to students. Hopefully, saving the teacher time and increasing the amount of open response questions given out.

5. ACKNOWLEDGEMENTS

I thank multiple NSF grants (e.g., 1931523, 1940236, 1917713, 1903304, 1822830, 1759229, 1724889, 1636782, 1535428, 1440753, 1316736, 1252297, 1109483, & DRL-1031398), the US Department of Education Institute for Education Sciences (e.g., IES R305A170137, R305A170243, R305A180401, R305A120125, R305A180401, & R305C100024) and the Graduate Assistance in Areas of National Need program (e.g., P200A180088 & P200A150306), and EIR the Office of Naval Research (N00014-18-1-2768 and other from ONR) and finally Schmidt Futures.

6. REFERENCES

- [1] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.
- [2] E. S. Charles, N. Lasry, S. Bhatnagar, R. Adams, K. Lenton, Y. Brouillette, M. Dugdale, C. Whittaker, and P. Jackson. Harnessing peer instruction in-and out-of class with mydalite. In *Education and Training in Optics and Photonics*, page 11143.89. Optical Society of America, 2019.
- [3] J. A. Erickson, A. F. Botelho, S. McAteer, A. Varatharaj, and N. T. Heffernan. The automated grading of student open responses in mathematics. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pages 615–624, 2020.
- [4] J. Gardner, C. Brooks, and R. Baker. Evaluating the fairness of predictive student models through slicing analysis. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 225–234, 2019.
- [5] N. T. Heffernan and C. L. Heffernan. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.
- [6] K. Y. Ku. Assessing students’ critical thinking performance: Urging for measurements using multi-response format. *Thinking skills and creativity*, 4(1):70–76, 2009.
- [7] M. E. Martinez. Cognition and the question of test item format. *Educational Psychologist*, 34(4):207–218, 1999.
- [8] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [9] J. Ocumpaugh, R. Baker, S. Gowda, N. Heffernan, and C. Heffernan. Population validity for educational data mining models: A case study in affect detection. *British Journal of Educational Technology*, 45(3):487–501, 2014.
- [10] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [11] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.
- [12] B. Riordan, A. Horbach, A. Cahill, T. Zesch, and C. M. Lee. Investigating neural architectures for short answer scoring. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 159–168, 2017.
- [13] J. Roschelle, M. Feng, R. F. Murphy, and C. A. Mason. Online mathematics homework increases student achievement. *AERA Open*, 2(4):2332858416673968, 2016.
- [14] M. G. Simkin and W. L. Kuechler. Multiple-choice tests and student understanding: What is the connection? *Decision Sciences Journal of Innovative Education*, 3(1):73–98, 2005.
- [15] J. Z. Sukkariéh and J. Blackmore. c-rater: Automatic content scoring for short constructed responses. In *Twenty-Second International FLAIRS Conference*, 2009.
- [16] J. Z. Sukkariéh, S. G. Pulman, and N. Raikes. Automarking: using computational linguistics to score short, free text responses. 2003.
- [17] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.
- [18] S. Zhao, Y. Zhang, X. Xiong, A. Botelho, and N. Heffernan. A memory-augmented neural model for automated grading. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 189–192. ACM, 2017.

Self-Regulated Learning and Science Reading of Middle School Students

Effat Farhana
North Carolina State
University
Raleigh, North Carolina, USA
efarhan@ncsu.edu

Teomara Rutherford
University of Delaware
Newark, Delaware, USA
teomara@udel.edu

Collin F. Lynch
North Carolina State
University
Raleigh, North Carolina, USA
cflynch@ncsu.edu

ABSTRACT

The role of self-regulated learning (SRL) behaviors for reading scientific texts has been largely recognized by researchers. Unfortunately, not all learners are effectively self-regulating. To provide effective support for SRL activities, it is necessary for us to understand how students adapt their self-regulation behaviors during reading. This study investigates students' SRL behaviors in science reading using historical data from a K-12 digital reading platform, Actively learn (AL). We analyze reading related SRL in four contexts, such as, domain-specific sequential pattern, question features, question and content difficulty, and teachers' interaction with the platform. We present findings of our work and seek advice on how the insight that we get from these findings can be used in our proposed methodology.

1. INTRODUCTION

Scientific literacy has been a central goal of international science education reforms for last decades, and researchers consider reading science texts as an integral part of science literacy [9]. Despite the importance of reading comprehension, students in the US lack reading proficiency. According to National Assessment of Educational Progress (NAEP) 2019 report, 37% 8th-graders in the US performed at or above NEAP reading proficiency level ¹ and this number is lower than that of 2017. An integral skill for reading is self-regulated learning (SRL) [14]. Unfortunately, the typical teacher/student ratio and teachers' priority for topic completion make it difficult for students to learn and practice reading skills and other SRL skills.

Digital reading platforms can provide opportunities to learn and practice SRL strategies in classroom settings. Retrospective analysis of rich data from digital platforms of can provide insight about students' learning pattern to support tailored interventions by instructors.

The present dissertation proposes four research questions

¹<https://nces.ed.gov/nationsreportcard/reading/>

Effat Farhana "Self-Regulated Learning and Science Reading of Middle-School Students" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 766 - 769

(RQs) to investigate students' reading and reading-related SRL behaviors within the AL platform ². The RQs are described as follows.

- **RQ1. [SRL Patterns and Performance Difference]** *How do students' score connect with their reading and SRL patterns?*
- **RQ2. [SRLs and Question Features]** *How do reading and SRL strategies vary with question features?*
- **RQ3. [SRLs and Content Difficulty]** *How do reading and SRL strategies vary with question and text difficulty?*
- **RQ4. [SRLs and Teachers' Interaction]** *How do teacher interactions with the system connect with students' reading and SRL behaviors?*

In the following subsections we present subquestions, motivations, and possible contributions associated with each RQ.

1.1 RQ1: SRL Patterns and Performance

We split the RQ1 into a subquestion as follows.

RQ1: Which reading and SRL patterns differ between high and low performing student?

RQ1.1 How reading patterns differ for science and social study?

The motivation of RQ1 is to identify reading and SRL behaviors for productive and unproductive students. Additionally, to understand how these behaviors vary for cross domain subjects. Findings of this RQ can be used to develop recommendation system targeted for specific group of students. Also, data driven analysis will be helpful for teacher to make tailored interventions for students.

1.2 RQ2: SRLs and Question Features

To conduct preliminary experiment, we split RQ 2 as follows:

RQ2: How do students' SRL strategies vary with question features?

RQ2.1 Does the association of SRL vary depending on question formats?

RQ2.2 How do other question feature: placement in the text,

²<https://www.activelylearn.com/>

length of question stem, standard usage are connected with students SRL behaviors?

The motivation of this RQ is twofold. First, to understand which question features prompted what types of SRLs? Second, to understand how question features predict performance in assignment score? While previous SRL researchers [1,5] focused mainly on question formats (i.e., multiple choice question, short answer, ...), our research will examine more fine grained question features to understand students' SRLs.

1.3 RQ3: SRLs and Content Difficulty

We will analyze RQ 3 into two phases as follows: *RQ3.1 How do SRL strategies vary with question difficulty?*

RQ3.2 How do SRL strategies vary with text complexity?

Previous two RQs do not distinguish difficulty level between question formats. RQ3.1 assesses question difficulty from student interaction data at *class* level. We compared our proposed approach with the IRT [10] approach.

1.4 RQ4: SRLs and Teachers' Interaction

The first three RQs analyze students SRL behavior considering their study pattern and question and text features. This RQ focuses on teachers usage the AL system and how it contributes to students SRL usage. We will focus on several teacher-behavior including: how frequently teachers are giving feedback and what question standards are they assigning to questions.

2. METHODS AND CURRENT PROGRESS

Currently, the analysis of the first research question, RQ1 is complete (accepted), a subquestion of the second research question, RQ2.1 is complete (accepted), and a subquestion of the third research question, RQ3.1 is under revision. As RQ3.1 is under submission, we present methodology and results of RQ1 and RQ2.1.

2.1 SRLs in AL

Our scope of this study is evaluating students' SRL usage in middle school science reading within the AL platform. AL reading assignments follow Next Generation Science Standards (NGSS) and have *text embedded* questions. Question formats can be multiple choice (MCQs) and short answer questions (SA) (i.e., fill in the blank and free texts). Questions are graded on a [0-4] scale. The platform's developers claim the platform promotes deep learning by close reading: annotating, highlighting, and engaging with text.

We identify three reading support features of AL as SRL: annotating [8], highlighting [13], and vocabulary lookups, as we believe these features serve as proxies for SRL behaviors. Science text involves concept words and vocabulary terms. Students' reading comprehension and motivation has been decreased due to introduction of concept words [4]. Vocabulary lookups help students to understand concepts when they come across new vocabularies. Annotating requires students comprehend text and write down in their own words [8]. Azevedo described taking notes, summarization, and reading notes in context of SRL strategies for science learning with hypermedia [2]. To select texts for highlighting, students monitor information and connect those to their prior knowledge [13].

2.2 Methodology of RQ1

We describe clustering approach followed by generating sequences, and applying differential sequence mining technique with 12,566 science and 16,240 social study student assignment data.

Clustering Students by Performance Score We calculated four types of scores for each MCQ and SA, resulting in eight performance features. These are: first attempt score, last attempt score, *Norm_last*, and *Long Submission*, *Norm_last* is the multiplication of last score by normalized attempts –the ratio of attempts a student's attempt to all students' attempts on that question in a class. *Long Submission* computes proportion of attempts a student made after the median time for all students on that question in a class. After observing the Silhouette width, we applied K-means clustering with K= 4 on both science and social study data .

Coding Student Actions Student activities in the AL are attempts on question answering and SRL. We coded following question answering first attempts of MCQ (M) and SA (S) and resubmissions of MCQ (m) and SA (s). SRL activities are a reading (R), annotating (A), a highlighting (H), and a vocabulary lookup (V). As the AL system does not record student sessions, we relied on a data-driven approach to identify sessions as described by Kovanovic et al. [7] and Adithya et al. [12]. We plotted histograms of time intervals between consecutive actions to identify last action of any time period. Based upon this analysis we chose a cutoff of 30 minutes as a *session* duration. We split all student activities within a single assignment by *session*. We compacted repeated events by + as done by Kinnenbrew et al. [6].

Frequent Patterns within Clusters Within each cluster, we applied the n-gram sequencing technique and include patterns containing at least one letter from the set {R, A, V, H}. Differential sequence mining algorithm [6], requires two parameters: s-support (frequency of a pattern *within* a cluster) and i-support (frequency of a pattern *within* one *action sequence*). We applied s-support = 0.5 to filter patterns exhibited by at least half of students within that cluster. Next, we applied the Kruskal-Wallis test to identify if a significant difference existed in the mean i-support value within the groups.

2.3 Methodology of RQ2

2.3.1 Methodology of RQ2.1

We used hierarchical linear models (HLMs) to model the relationship between observed behaviors and performance, with assignment at level one, nested within students (level two), nested within classes (level three). We built three models for three different response variables: overall assignment score, MCQ score, and SA score. The fixed-effect variables were the SRL features and number of questions in assignment; these variables were at *Level 1*. Assignment, student, and class were all modeled as random intercepts.

3. RESULTS

In this section we present results of RQ 1 and RQ 2.1.

3.1 Results of RQ1

Four resulting science clusters with student counts (n) were: SA_sc (SA performers in science, n = 4,474), MC_sc (MCQ performers in science, n= 3114), L_sc (low performers in science, n =2,363), and H_sc (high performer in science, n =

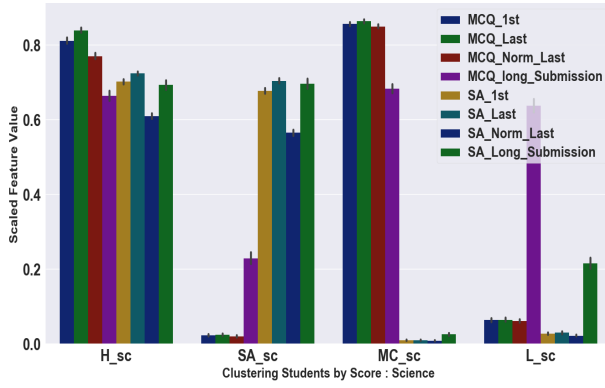


Figure 1: Science Student Clustering by Score

2,636). Similar as in science student clustering, we observe four different groups in social studies: H_{ss} ($n = 8,948$), L_{ss} ($n = 2760$), MC_{ss} ($n = 2928$), and SA_{ss} (1604). We focused primarily identifying high and low performing student behaviors.

Science Cluster Analysis: Considering H_{sc} vs L_{sc} group, two more frequently used patterns describing SA answering behaviors by H_{sc} students were RS (I-sup_Diff = 0.17, $p < 0.001$) and RS+ (I-sup_Diff = 0.08, $p < 0.001$). RS and RS+ describe reading prior attempting one (S) or multiple (S+) SAs. Thus, reading prior SA attempt were linked to high performances. H_{sc} group students also exhibited more annotation behavior than L_{sc} students (I-sup_Diff = 0.03, $p < 0.001$). Three MCQ attempt related patterns were more exhibited by L_{sc} group students: RM (I-sup_Diff = -0.16, $p < 0.001$), and V+M (I-sup_Diff = -0.001, $p < 0.05$), and RH+M (I-sup_Diff = -0.002, $p < 0.001$). From Figure 1, we observe L_{sc} group students have more MCQ_Long_submissions and lower MCQ_Last scores. We conclude L_{sc} group students struggled in choosing the correct MCQ option.

Social Study Cluster Analysis: Our analysis showed higher-performing students in social study assignments read more frequently before attempting SA and MCQs. Additionally, they looked up more vocabulary. In contrast, low performing students read after attempting SAs. They also had higher resubmission rate of SA questions followed by read event. Our observed patterns explain the way high and low performing students navigated the SA questions. We conclude reading and looking up vocabularies for comprehending the concept prior answering a SA led to score differences for social study subject.

Differential Sequence Mining: Science vs Social Study:

We begin with our results for the H_{sc} vs H_{ss} comparison. Science students exhibited reading behavior *after* SA submissions compared to social studies: SR (I-sup_Diff = 0.16, $p < 0.001$), S+R (I-sup_Diff = 0.12, $p < 0.001$). Examining the descriptive statistics, we noticed the mean SA score is higher in social study assignments (SA First = 2.56, SA Last = 2.62) compared to science (SA First = 2.46, SA Last = 2.58) ones. Additionally, mean MCQ scores of science is higher (MCQ first = 2.80, MCQ Last = 2.89) than those of social study (MCQ First = 2.17, MCQ Last = 2.19). Thus, we compared MC_{sc} vs MC_{ss} and SA_{sc} vs SA_{ss} group. The relatively lower mean SA score in sci-

ence can be explained by SR (I-sup_Diff = 0.16, $p < 0.001$) and S+R (I-sup_Diff = 0.14, $p < 0.001$). Analyzing MC_{sc} vs MC_{ss} group, students with science assignments exhibited more reading behavior before attempting MCQ as described by pattern R+M (I-sup_Diff = 0.0192, $p < 0.001$). Although the two subject domains are different, our analysis shows reading prior attempting a question associated with higher score in both domains.

3.2 Results of RQ2.1

Table 1: Results from HLM Measuring Association between SRL and Science Score

<i>L1 Level</i> (Assignment)	β	<i>B</i>	<i>SE</i>	<i>p</i>
Overall Score				
Intercept		6.533	0.402	<0.001
A	0.055	0.582	0.062	<0.001
H	0.028	0.492	0.072	<0.001
V	0.021	0.275	0.055	<0.001
MCQ Score				
Intercept		5.510	0.369	<0.001
A	0.024	0.206	0.038	<0.001
H	0.016	0.228	0.045	<0.001
V	-0.003	-0.036	0.031	0.259
SA Score				
Intercept		1.699	0.232	<0.001
A	0.040	0.271	0.038	<0.001
H	0.019	0.210	0.043	<0.001
V	0.036	0.289	0.035	<0.001

We report standardized effect size using the formula $\beta = (B * SD_x) / SD_y$ (see e.g., [11]). Table 1 presents our findings. All SRL-related variables had positive and statistically significant association with overall science score. Considering question format, the predictive power of note taking was highest ($B = 0.271$, $\beta = 0.041$, $p < 0.001$) followed by highlighting ($B = 0.210$, $\beta = 0.019$, $p < 0.001$), and vocabulary lookups ($B = 0.289$, $\beta = 0.036$, $p < 0.001$). Considering MCQ format, all but the vocabulary lookups continued to be statistically significant positive predictors of MCQ score.

4. FUTURE WORK AND ADVICE SOUGHT

Proposed Methodology of RQ 2.2 We will use multi-task learning to predict common SRL behavior of students for each question (considering question features) and performance on the question. Thus, we will be able to identify students who need help.

Proposed Methodology of RQ 3.2 We will analyze text readability and complexity including lexical, semantic, and argumentation of the text and SRL usage. To analyze readability of science text, we will examine Coh-Metrix [3] and Python's readability package³. Additionally, we will examine the argumentation analysis in SA response, particularly questions asking for reasoning, e.g. *Why*, *How*, and *Explain*.

Proposed Methodology of RQ 4 To answer RQ 4, we will perform exploratory analysis to answer the sub questions and calculate association with students' SRL behaviors.

A key limitation of our analysis is, we do not know many confounding variables such as, how teachers used AL assignments (in-class, homework assignment, or extra reading), demographic of students, and how they were using SRLs (i.e., teacher might instruct to take notes). Thus, we seek advice on following aspects:

³<https://pypi.org/project/readability/>

- Is our proposed method of RQ 2.2 generalizable to other context, considering the limitation of our study? The motivation of RQ 2.2 is to provide data-driven recommendation to researchers and educators.
- Beyond my proposed methodology, what other analysis could be more beneficial to understand students' SRL strategies in science reading?

5. REFERENCES

- [1] S. Agrawal, G. R. Norman, and K. W. Eva. Influences on medical students' self-regulated learning after test completion. *Medical education*, 46(3):326–335, 2012.
- [2] R. Azevedo. The role of self-regulated learning about science with hypermedia. *Recent innovations in educational technology that facilitate student learning*, pages 127–156, 2008.
- [3] S. A. Crossley, K. Kyle, and D. S. McNamara. The tool for the automatic analysis of text cohesion (taaco): Automatic assessment of local, global, and text cohesion. *Behavior research methods*, 48(4):1227–1237, 2016.
- [4] Y.-S. Hsu, M.-H. Yen, W.-H. Chang, C.-Y. Wang, and S. Chen. Content analysis of 1998–2012 empirical studies in science reading using a self-regulated learning lens. *International Journal of Science and Mathematics Education*, 14(1):1–27, 2016.
- [5] S. Jordan. Student engagement with assessment and feedback: some lessons from short-answer free-text e-assessment questions. *Computers & Education*, 58(2):818–834, 2012.
- [6] J. S. Kinnebrew and G. Biswas. Identifying learning behaviors by contextualizing differential sequence mining with action features and performance evolution. *International Educational Data Mining Society*, 2012.
- [7] V. Kovanović, D. Gašević, S. Dawson, S. Joksimović, R. S. Baker, and M. Hatala. Penetrating the black box of time-on-task estimation. In *Proceedings of the fifth international conference on learning analytics and knowledge*, pages 184–193, 2015.
- [8] T. Makany, J. Kemp, and I. E. Dror. Optimising the use of note-taking as an external cognitive aid for increasing learning. *British Journal of Educational Technology*, 40(4):619–635, 2009.
- [9] S. P. Norris and L. M. Phillips. How literacy in its fundamental sense is central to scientific literacy. *Science education*, 87(2):224–240, 2003.
- [10] G. Rasch. Studies in mathematical psychology: I. probabilistic models for some intelligence and attainment tests. 1960.
- [11] T. Rutherford, J. J. Long, and G. Farkas. Teacher value for professional development, self-efficacy, and student outcomes within a digital mathematics intervention. *Contemporary educational psychology*, 51:22–36, 2017.
- [12] A. Sheshadri, N. Gitinabard, C. F. Lynch, T. Barnes, and S. Heckman. Predicting student performance based on online study habits: A study of blended courses. *International Educational Data Mining Society*, 2018.
- [13] P. H. Winne, J. C. Nesbit, I. Ram, Z. Marzouk, J. Vytasek, D. Samadi, and J. Stewart. Tracing metacognition by highlighting and tagging to predict recall and transfer. *AERA Online Paper Repository*, 2017.
- [14] B. J. Zimmerman. Self-regulated learning and academic achievement: An overview. *Educational psychologist*, 25(1):3–17, 1990.

Developing Curriculum Analytics and Student Social Networking for Graduate Employability Model

Aleksandr Gromov
University of Technology Sydney
aleksandr.d.gromov@student.uts.edu.au

ABSTRACT

This research focuses on developing graduate employability among university students. The ability to find a graduate position became one of the key tertiary education goals for enrolled students. However, there are lots of factors that affect graduate employability. At the same time, students could be unaware of employability complexity, and their choices may be made blindly. I aim to create a graduate employability model that will help to build a learning path and strategy to the desired career. I am using curriculum profiling, and student performance data to model skills and abilities students develop in their subjects. Besides, I am building a student social network model to analyse students' interactions and ties. Ultimately, my research aims to predict graduate employment and recommend options for better student choices.

Keywords

Graduate employability, curriculum analytics, network analysis

1. INTRODUCTION

Graduate employability became one of the key indicators of university performance. Despite the desire to be standalone institutes and the fact that university education is much broader than simple skill training, universities accepted employability development as one of the goals for tertiary education to satisfy student and industry needs. For instance, graduate attributes, derived from professional industry requirements, are injected into the curriculum, and work-integrated learning became a part of the learning process, aimed at providing work-related experience to students. However, after completing the course, students are not equally employable; one of them find a relevant position upon graduation, while others are stuck without any job offers. What makes one graduate more employable than another?

Literature reveals different factors that affect graduate em-

ployability [1, 8]. They can be aggregated as social, human, behavioural and environmental factors [4]. Social factors define the position of a graduate person in society. As the result, attending a better university, having a large network, belonging to certain social classes will benefit employability chances. In addition, human factors describe personal traits a person have. So, skills developed during the learning course and previous work experience will improve graduate's employability in comparison with another graduate, who is missing these abilities and practice [5]. Furthermore, behavioural factors combine one's attitude toward successful employability [3]. For example, being an active job seeker and dedicated participant of career-related events and workshops makes a difference with a passive waiting for a good position on market demand. Finally, environmental factors are not related to a graduate, but the market situation in general [8]. Economics recession has a negative impact on employability in general, without regards to any personal factors. However, mentioned facets of graduate employability relatively objective and can be analysed by data-driven approaches [2]. There is another, subjective, dimension of perceived employability, which effects chances to be employed based on individual self-evaluation and believes [8].

In my research, I aim to create a student or graduate employability model. However, I understand the complexity of all factors. Moreover, the nature of some factors, such as the economic situation, cannot be altered on a graduate or even university level, falling into the mercy of global processes. Thus, I decided to focus on skills and competences as human factors, and student networks as social factors for my research.

2. GRADUATE EMPLOYABILITY MODEL

In my research, I focus on creating a graduate employability model and investigating the effect of skills, developed by students at university, and student social networks, build through various subjects and courses, on employability after graduation.

2.1 Developing skills through the degree

Students are required to undertake a number of credit points to obtain the degree, which is done by completing multiple subjects. At the same time students are developing their skills and abilities, going through various tasks, assignments and group activities. Knowledge and skill development are integrated into the curriculum and it makes curriculum the source for skill data mining. While curriculum data shows

Aleksandr Gromov "Developing Curriculum Analytics and Student Social Networking for Graduate Employability Model" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 770 - 772

quantitative skill outcomes, students performance will be used for quality research. Clearly, students with in-depth, comprehensive approach will benefit more from the subjects in comparison with effortless students. Implementing these aspects of skill development is part of my project.

2.1.1 Current work

Recently, I created a curriculum profile for the university. Curriculum profile is a hierarchical data structure with skills as the basis nodes, which are aggregated into larger nodes, such as subjects, courses, degrees, faculties or the whole university. Course coordinators benefit from curriculum profile by visualising course outcomes and comparing intended expectations with reality. Students will be able to make more rational choices selecting from the various subjects and navigate their degree with more predictable results.

The curriculum profile is build using off-the-shelf ontology and automated data curriculum data collection. However, the idea behind curriculum profiling requires modular structure with replaceable components. Thus, any ontology that works according to aligned rules can be used as a data source for the curriculum profile via application programming interfaces (APIs).

2.1.2 Future work

Firstly, current curriculum profile is working for one university and has only two modes: the whole university and three selected data science courses. Future development will include all courses from the university available for analysis and comparison.

Secondly, curriculum data is miscellaneous and exist in multiple forms. On the one hand, different universities or even faculties have incompatible curriculum data. On the other hand, some part of the curriculum, such as “Teaching Strategies” or “References”, are less meaningful for text analytics, while others, such as “Content” or “Learning Outcomes”, are richer. The planned research aims to compare each part of the curriculum data to reshape and refine a data source for better text analytics. In addition, it eases the compatibility issues between faculties and universities.

Finally, the created curriculum profile will be matched with student performance data to provide quality perspective on developed skills and market data of employed graduates. It will allow visualising learning path leading to the successful employment, revealing key subjects and skills that helped to achieve it in comparison with other graduates.

2.2 Student networks

Another part of graduate employability model is student social networks. During the learning process, students are involved in multiple subjects. Over the years of study, they interact with hundreds of students, tutors, industry representatives [6, 7]. Even more, indirectly, they can know thousands of other students via the people they know. The process of forming these networks is random. However, networks are reported as an important factor that affects employability [2], and student social networks are a great source of strong and weak ties useful for graduate employability. In my research, I aim to model student networks and predict

how they change the ability to become employed after graduation.

2.2.1 Future work

As part of network analysis, I plan to create a bipartite university network by semester for a selected period of time. The nodes of the network will be students, and edges will be subjects they have selected. Overall network visualisation will help to understand student relationships, identify key subjects and dynamics of network spread. The finalised network will be compared with career data from graduates, who were part of the network as students, to identify choices they made in networking and career outcomes. The contribution of this study will be a student networking model that can predict employment chances for a given student and recommend networking strategies to become more employable.

3. ADVICE SOUGHT

For this doctoral consortium, I seek for advice regarding two questions. Firstly, *what mathematical, probabilistic and statistical methods could benefit my curriculum and network analysis*. I identified several common methods used for other studies. So, skills can be presented as vectors, and further comparison will be reduced to vector comparison, metrics, and space projections. Similarly, I adopted networking methods that allow evaluating network density, clustering, diameter and reach. However, I am looking for more models and methods for my curriculum and network analysis.

Secondly, *are there other factors that affect graduate employability and can be improved at university*. Currently, I use the curriculum data for extracting skills and related careers to identify possible outcomes after completing a subject or course. After that, I will use student performance data to normalise skills outcomes. Also, I use student enrolment data to build bipartite networks. After that, all this results will be matched against actual employment data after graduation. My method creates investigates learning paths and strategies that could lead to successful employment. Thus, I consider human capital (personal skills and abilities) and social factor (student networks), as factors of graduate employability to be improved through the degree. However, I acknowledge the complexity of other factors and their interactions. My research will benefit from experts opinion on developing graduate employability at universities.

4. REFERENCES

- [1] E. Berntson and S. Marklund. The relationship between perceived employability and subsequent health. *Work & Stress*, 21(3):279–292, 2007.
- [2] S. Biancani and D. A. McFarland. Social networks research in higher education. In *Higher education: Handbook of theory and research*, pages 151–215. Springer, 2013.
- [3] B. V. Carolan. *Social network analysis and education: Theory, methods & applications*. Sage Publications, 2013.
- [4] M. Clarke. Rethinking graduate employability: The role of capital, individual attributes and context. *Studies in Higher Education*, 43(11):1923–1937, 2018.
- [5] B. Freudenberg, M. Brimble, and C. Cameron. Wil and generic skill development: The development of business

students' generic skills through work-integrated learning. *Asia-Pacific Journal of cooperative education*, 12(2):79–93, 2011.

- [6] U. Israel, B. P. Koester, and T. A. McKay. Campus connections: Student and course networks in higher education. *Innovative Higher Education*, pages 1–17, 2020.
- [7] M. Newman. *Networks*. Oxford university press, 2018.
- [8] D. Vanhercke, N. De Cuyper, E. Peeters, and H. De Witte. Defining perceived employability: a psychological approach. *Personnel Review*, 2014.

Overcoming Foreign Language Anxiety in an Emotionally Intelligent Tutoring System

Daneih Ismail
DePaul University
dismail1@depaul.edu

ABSTRACT

The interactions between learning and emotions are bidirectional. Positive emotions such as motivation, engagement, and happiness induce learning gain. Negative emotions such as anxiety, confusion, and frustration weaken learning achievements. Understanding the learner's mental and emotional state would promote their positive emotion and diminish their negative emotion, which in return, increases learning acquisition. One of the most negative emotions that affect foreign language learning is anxiety. Through our study, we would like to investigate how to detect foreign language anxiety (FLA) then how to reduce and eventually overcome FLA. In the context of FLA, we propose a sensor-free anxiety detector. To overcome FLA, we propose a pedagogical animated agent that provides emotional support. Our preliminary findings showed that a pre-test of a Foreign Language Classroom Anxiety Scale (FLCAS) is effective to predict FLA in the context of an e-learning system.

Keywords

Foreign language anxiety, Emotion, Affect, Intelligent Tutoring System, Sensor-free, Animated Agent

1. INTRODUCTION

Learning and emotions are interrelated. The brain architecture allows complex interactions between emotion and cognition. The brain region work in the integration of the emotional and cognitive process that impact behaviors [18]. A positive, supportive learning environment can escalate positive emotions, which in return, can increase learning gains. On the other hand, a negative learning environment could increase negative emotions, which would weaken learning achievement [9]. Learning a foreign language is challenging because of the cognitive, emotional, and native language proficiency [14]. Anxiety plays critical role in reducing foreign language acquisition [19]. There are several reasons that induce Foreign Language Anxiety (FLA) such as fear of neg-

ative evaluation, communication apprehension, test anxiety [8], task complexity [12], and lack of emotional intelligence [20]. FLA impacts the learner's production and retention [19]. Moreover, FLA produces unwillingness to communicate in the foreign language [15, 17] and reduces the motivation to learn [16]. Furthermore, it divides attention between emotion and cognition which makes performance less efficient [11].

To measure FLA, researchers have used physical measurements [9], self-report [8], and facial recognition [7].

To overcome FLA, researchers have used ITSs [13], robots [3], or games [21]. Each study employs different strategies such as animated agents that provide communicating strategies and affective backchannels [5], soothing music [13], or adjusting the difficulty to suit the learner's level [1, 4, 6, 13].

In our research, we are focusing on foreign language anxiety (FLA). We would like to build a sensor-free emotionally intelligent tutoring system that reduces and eventually overcomes FLA. To achieve our goal we need to understand the causes of FLA, to detect FLA, and to provide interventions that overcome FLA.

The first research question is how to detect the student's anxiety level in an e-learning system. Based on [8], three main reasons produce FLA; fear of negative evaluation, communication apprehension, and test anxiety. In previous studies, we used sensor-lite approach which uses minimal sensors like self-report. We analyzed language difficulty self-report, system difficulty self-report, score of exercise, and pre-test of FLCAS to predict anxiety level [9]. However, for future studies, we would like to investigate sensor-free approach to avoid asking the learner. Consequently, we hypothesized that a pre-test of FLCAS which consists of these three parts would be effective for predicting FLA in the context of an e-learning system.

The second research question is what is the best intervention to reduce FLA in an e-learning system. According to [2], the experimental condition which included animated agents showed positive effects on reducing language barriers while the control group showed shyness and worry when learning Russian as a foreign language. Providing emotional support reduces anxiety [10]. Consequently, we hypothesized that animated agents that provide emotional support are effective for reducing FLA.

Daneih Ismail "Overcoming Foreign Language Anxiety in an Emotionally Intelligent Tutoring System" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 773 - 775

2. PROPOSED CONTRIBUTIONS

Our project is significant because we will use a sensor-free emotionally intelligent tutoring system that overcomes anxiety when learning English as a second language. Using a sensor-free approach will allow the learner to use the product in any environment without the interruptions of a physical sensor. Also, it will be capable of recognizing FLA without asking the learner about their feeling. The ITS will be able to identify learner's anxiety levels and provides adequate support.

The proposed project is unique because it uses animated agents that provide emotional support to reduce FLA. The benefits of this work are decreasing and eventually overcoming FLA. It will help foreign language learners defeat their negative emotions. Moreover, it will generate a relaxing, encouraging, and motivating learning environment which ultimately improves the learning gain.

3. RESULTS SO FAR

3.1 Previous Studies

We did an experiment to identify FLA. 30 participants who are non-native English speakers join the study. They completed FLCAS and demographic information. Then, they answered 27 exercises in grammar, vocabulary, speaking, and listening. We did a correlation analysis to understand the relationship between physical measurements and level of anxiety self-report. We found a significant positive correlation between level of anxiety self-report, blood pressure, heart rate, and eye fixation. Also, we identified FLA by analyzing interaction of learners with e-learning system. We found that time on task, and number of mouse clicks were not significant. While language difficulty self-report, system difficulty self-report, and score of exercise were effective to predict FLA in context of e-learning system [9].

We did other analyses to predict FLA in the context of e-learning system. We predicted FLA based on subject and regardless of type of exercise using pre-test FLCAS components. Depending on exercise type, we used sensor-free prediction using various components of FLCAS. For example, average communication apprehension score was 40% effective to predict FLA in context of listening exercise. Grammar and vocabulary predictions were not significant. For overall FLA, we predicted that average fear of negative evaluation, average communication apprehension, language difficulty self-report, system difficulty self-report, and exercise score account for about 43% of variation in anxiety. We used sensor-lite in this prediction by using language and system difficulty self-report to increase accuracy of prediction.

3.2 Current Study

There will be 180 participants randomly assigned to six groups. They should be non-native English speakers and non-fluent. Their age should be above 18 years old.

The participants start the study by answering some demographic information (native language, age, educational level, and English level). Then they complete FLCAS. After that, they are assigned to one of the six groups (control, textbase supportive feedback, voice supportive feedback, voice feedback, agent supportive feedback, or agent feedback). All the

six groups have the same material which teaches and provide practice in English listening, vocabulary, grammar, writing, and reading. The difference between the groups is in the way feedback is provided. The participants are expected to learn the material then do 20 exercises. After each exercise, there is a self-report that includes language difficulty, system difficulty, and anxiety level. When the participant finishes all the exercises and self-report we send them a \$20 Amazon e-gift card.

For the first research question, we will do a statistical analysis to understand the relationship between FLCAS and learner's current level of anxiety when using an e-learning system. We would like to predict the learner's current level of anxiety using three main components of FLCAS: communication apprehension, fear of negative evaluation, and test anxiety [8]. We will use regression and 10-fold cross-validation to verify the results. For the second research question, we will use Mann Whitney U test to understand which intervention is effective to reduce anxiety.

Then we will use the data from both research questions to build an ITS that reduces FLA.

4. ADVICE SOUGHT

There are two main aspects of research on which advice is sought. First, the set of features used to predict FLA using a sensor-free approach. Our preliminary study showed that using sensor-lite is effective to predict FLA. Sensor-lite uses language difficulty, system difficulty, score, and pre-test of FLCAS as predictors. We would like to use sensor-free without having the language and system difficulty self-report but the model fit drops from 40% to 20%.

Second, algorithms, tools, and applications to build an English foreign language intelligent tutoring system which reduces FLA. So far, we built an e-learning system but we want to upgrade it to be intelligent tutoring system. We tried using CTAT as platform but it was not compatible with the animated agent application we are using. We would like to find the best practices to build the ITS.

5. CONCLUSIONS AND FUTURE WORK

Foreign language anxiety is a major obstacle to learning a foreign language. Identifying FLA then reducing it and eventually overcoming it is a novel approach to improving foreign language acquisition. Using sensor-free anxiety detector and altering the system to reduce anxiety is a promising approach. Through our study, we hope we can predict anxiety using sensor-free approach and reduce anxiety using emotional supportive agent.

6. ACKNOWLEDGMENTS

This experiments was partially supported by a grant from College of Computing and Digital Media at DePaul University. And Media Semantics provided us with a free license for the animated agent.

7. REFERENCES

- [1] M. J. Abu Ghali, A. Abu Ayyad, S. S. Abu-Naser, and M. Abu Laban. An intelligent tutoring system for

- teaching english grammar. *International Journal of Academic Engineering Research (IJAER)*, 2018.
- [2] A. Al-Kaisi, A. Arkhangelskaya, O. Rudenko-Morgun, and E. Lopanova. Pedagogical agents in teaching language: Types and implementation opportunities. *International E-Journal of Advances in Education*, 5(15):275–285, 2020.
 - [3] M. Alemi, A. Meghdari, and M. Ghazisaedy. The effect of employing humanoid robots for teaching english on students’ anxiety and attitude. In *2014 Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, pages 754–759. IEEE, 2014.
 - [4] M. I. Alhabbash, A. O. Mahdi, and S. S. A. Naser. An intelligent tutoring system for teaching grammar english tenses. *European Academic Research*, 2016.
 - [5] E. Ayedoun, Y. Hayashi, and K. Seta. Adding communicative and affective strategies to an embodied conversational agent to enhance second language learners’ willingness to communicate. *International Journal of Artificial Intelligence in Education*, 29(1):29–57, 2019-03.
 - [6] C.-M. Chen and T.-H. Lee. Emotion recognition and communication for reducing second-language speaking anxiety in a web-based one-to-one synchronous learning environment. *British Journal of Educational Technology*, 42(3):417–440, 2011.
 - [7] Y. Guo, J. Xu, and X. Liu. English language learners’ use of self-regulatory strategies for foreign language anxiety in china. *System*, 76:49–61, 2018.
 - [8] E. K. Horwitz, M. B. Horwitz, and J. Cope. Foreign language classroom anxiety. *The Modern language journal*, 70(2):125–132, 1986.
 - [9] D. Ismail and P. Hastings. Identifying anxiety when learning a second language using e-learning system. In *Proceedings of the 2019 Conference On Interfaces and Human Computer Interaction*, pages 131–140, 2019.
 - [10] Y. X. Jin and J.-M. Dewaele. The effect of positive orientation and perceived social support on foreign language classroom anxiety. *System*, 74:149–157, 2018.
 - [11] Z. Kralova and G. Petrova. Causes and consequences of foreign language anxiety. *XLinguae*, 10(3):110–122, 2017.
 - [12] E. E. Levitt. *The psychology of anxiety*. Routledge, 2015.
 - [13] H.-C. K. Lin, C.-J. Chao, and T.-C. Huang. From a perspective on foreign language learning anxiety to develop an affective tutoring system. *Educational Technology Research and Development*, 63(5):727–747, 2015.
 - [14] H.-j. Liu. Understanding efl undergraduate anxiety in relation to motivation, autonomy, and language proficiency. *Electronic Journal of Foreign Language Teaching*, 9(1), 2012.
 - [15] M. Liu. Anxiety in efl classrooms: Causes and consequences. *TESL Reporter*, 39(1):13–32, 2006.
 - [16] M. Liu and W. Huang. An exploration of foreign language anxiety and english learning motivation. *Education Research International*, 2011, 2011.
 - [17] M. Liu and J. Jackson. An exploration of chinese efl learners’ unwillingness to communicate and foreign language anxiety. *The Modern Language Journal*, 92(1):71–86, 2008.
 - [18] L. Pessoa, L. Medina, P. R. Hof, and E. Desfilis. Neural architecture of the vertebrate brain: implications for the interaction between emotion and cognition. *Neuroscience & Biobehavioral Reviews*, 2019.
 - [19] S. H. Rafada, A. A. Madini, et al. Effective solutions for reducing saudi learners’ speaking anxiety in efl classrooms. *Arab World English Journal (AWEJ)*, 2017.
 - [20] K. Shao, W. Yu, and Z. Ji. An exploration of chinese efl students’ emotional intelligence and foreign language anxiety. *The Modern Language Journal*, 97(4):917–929, 2013.
 - [21] J. C. Yang and B. Quadir. Effects of prior knowledge on learning performance and anxiety in an english learning online role-playing game. *Journal of Educational Technology & Society*, 21(3):174–185, 2018.

The Effect of Visual Cues on Cognitive Load Depending on Self-Regulation in Video-Based Learning

Kakyeong Kim
Seoul National University
bettybetty3k@gmail.com

Il-Hyun Jo
Ewha Womans University
ijo@ewha.ac.kr

ABSTRACT

Recently, online learning has been increasingly used due to its advantages that allow people to study anytime and anywhere. Learners, on the other hand, are separated from the instructor in video-based learning, which makes learners difficult to maintain their motivation until the end of the activity. Therefore, it is required to provide instructional treatment so that learners keep their motivation and be immersed in learning. Visual cues, lowering cognitive load, are known a putative way in which learners can distinguish essential information from irrelevant one. The aim of this study is to explore the effect of visual cues on cognitive load depending on the level of self-regulation. The result shows that self-regulation lower cognitive load in the non-visual-cue group as time series, but self-regulation doesn't have an effect on cognitive load in the visual-cue group. This indicates learners in the non-visual-cue group experience difficulty to keep their motivation so that they are hard to put mental effort into the learning. This study suggests that the pupil dilation which reflects cognitive load can be predicted by behavior log which indicates self-regulation. Therefore, it is necessary to enhance learners' cognitive strategies, as well as the reduction of the factors causing unnecessary cognitive load.

Keywords

Visual Cues, Cognitive Load, Self-Regulation, Video-Based Learning, Learning Analytics

1. INTRODUCTION

With the development of technology, online learning has been dramatically increased and people can easily watch videos from different platforms on diverse terminals, such as desktop, tablet, phone [16]. It allows learners to study anytime and anywhere [6]. Learners, on the other hand, are separated from the instructor in video-based learning, which makes learners difficult to maintain their motivation until the end of the activity. Therefore, it is required to provide instructional treatment so that learners can keep motivation and be immersed in learning.

According to Mayer, meaningful learning requires learner to select relevant information, organize the information into coherent representation and integrate this representation into existing knowledge [11]. Multimedia instructions are effective when it is designed in accordance with how human mind works [7].

Kakyeong Kim and Il-Hyun Jo "The Effect of Visual Cues on Cognitive Load Depending on Self-Regulation in Video-Based Learning" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 776 - 780

Working memory has limited capacity when dealing with novel or unorganized information, since it needs more cognitive processing [17]. Cognitive load theory (CLT) is based on the concept that people have a limited working memory and processing capacity [23]. Therefore, the ease of information processing in working memory is main interest of cognitive load theory [13].

CLT consists of three elements: intrinsic load, extraneous load, and germane load [12]. Among them, extraneous load and germane load are affected by instructional design. Extraneous load is risen when unnecessary cognitive processing is needed. Germane load is decreased when learners are not involved in deep cognitive processing, such as organizing or relating the material to prior knowledge [15]. Relatedly, using cues when designing instruction can draw learners' attention to essential elements in learning. Therefore, providing visual cues can lower extraneous load and increase germane load [8].

The relationship between cognitive load and self-regulation sheds light on when and why learners adopt their behaviors and how these behavioral changes are related with cognitive load [5]. Self-regulated learning (SRL) refers to learning with student's goal-directed self-generated thoughts, feelings, strategies, and behaviors [14]. A self-regulated learner plans, monitors, reflects, and adjusts his/her learning process metacognitively, so he/she shows self-paced learning behaviors in computer-mediated learning environment [19, 24]. Learners' regulating behaviors can be assessed as how frequently learners use the learning strategy with the Learning Management Systems (LMSs) in real learning time [4, 18]. Therefore, learners' online behaviors in an e-learning player can be used to analyze self-regulation.

In sum, two hypotheses are created. Firstly, visual cues would be a factor to promote germane load with decreasing extraneous load in time series, which is measured by pupil dilation. Secondly, self-regulation would have a moderate effect on learners' cognitive load. This study aims to explore the effect of the instructional design with visual cues in regard with learners' cognitive load depending on the level of self-regulation.

2. METHODOLOGY

2.1 Participants

Participants were recruited through online notices for a month. Since an ophthalmic disease influences the eye-tracking, participants were asked if they suffered from any of it. A total 100 undergraduate students (46 female, $M = 24$, $SD = 1.79$) took part in the study on a voluntary basis. Then they were randomly assigned to the group with or without visual cues. Among 100 participants, 23 participants were excluded from the analysis due to mechanical faults (e.g., calibration was cancelled or failed, recording was stopped). In the end, 77 participants' data were used and analyzed.

2.2 Procedures

Table 1. Research procedures

Recruitment		Online Notices (Korean Undergraduates)
Screening		Based on Eyesight, Major, Sex
Orientation (5 min)		Explanation about Research and IRB approval, Informed Consent
Experiment (80 min)	Questionnaire (10 min)	Motivated Strategies for Learning Questionnaire (MSLQ)
	Pre-Test (25 min)	6 Multiple-choice Items (from PSAT)
	Video-Based Learning (17 min)	Learning Material for PSAT Problem Solving Strategies
	Post-Test (25 min)	6 Multiple-choice Items (from PSAT)
	Questionnaire (3 min)	The ITC-Sense of Presence Inventory (ITC-SOPI)
Interview (60 min)		Interview with Eye Movement Data

The experiment proceeded in five phases (Table 1). Before starting the experiment, each participant took a 5-minute instruction about the procedure. All the phases were carried by computer. In the first stage, participants responded to Motivated Strategies for Learning Questionnaire (MSLQ). In the second stage, with eye calibration for the eye tracking, a 25-minute pre-test were administered to assess participants' prior knowledge. In the third stage, participants watched the 17-minute video lecture. The video lectures had been designed as two versions whether providing visual cues or not. In the meantime, participants' eye movement and behavior log were recorded with the e-learning player. In the fourth stage, participants took a 25-minute post-test for measuring learning achievement. Finally, they filled out the ITC-Sense of Presence Inventory (ITC-SOPI) scale and then they were interviewed for the comparison between the eye movement and the subjective learning experience response.

2.3 Learning Materials

Learning materials for Public Service Aptitude Test (PSAT) problem solving published by online distance educational institution [26] were used after having been edited. In South Korea, PSAT was devised to test the public officer applicants how well they deal with the public service. For the purpose of this research, the 'data interpretation ability' section in PSAT was only used. This material consists of four problem solving items and total learning time is about 17 minutes.

Two versions of the learning materials were developed with visual cues or without them. Both versions of the learning materials have an illustrated document and a spoken explanation. The experimental group was provided additional colored visual cues when an instructor explains or emphasizes the learning content. That is, colored visual scribbles are added in real time by the instructor while he/she speaks. By contrast, the visual cues were not given to the control group.

2.4 Measures

2.4.1 Physiological measure of cognitive load

Pupillary response measures people's cognitive processing load as a physiological measure [25]. Especially, pupil dilation reflects capacity utilization and relates to cognitive demand [1]. Mean

pupil dilation is a useful for measuring cognitive load [2, 13]. Therefore, mean pupil dilation is measured to analyze participants' mental effort and cognitive load during test and learning time. A Tobii Pro X2-30 eye-tracker and Tobii Studio software was operated at a sampling rate of 30 Hz. Pupillary response was calibrated to the environmental brightness and display luminance for controlling external noise.

2.4.2 Behavioral measures of self-regulation

In order to measure and analyze learners' self-regulating behavior, behavioral log data were collected via e-learning player automatically (figure 1). The learning-related behavior was counted to assess learners' self-regulation [3]. In this study, self-regulation is defined as the sum of frequencies corresponding to every operation to regulate learners' learning and learning environment. Therefore, the frequencies of play, pause, skip, replay, volume change, and rate change were used to analyze learners' self-regulation. The e-learning player was developed by 4CSoft and EduTech Convergence Lab in South Korea.

2.4.3 Learning achievement

Learning achievement was measured as the difference between pre-test scores and post-test scores. Pre-test and post-test were designed based on PSAT. Each test material consists of 6 multiple choice items. Both pre-test and post-test were verified by PSAT subject matter expert.

2.4.4 Questionnaires

2.4.4.1 Motivated Strategies for Learning Questionnaire (MSLQ)

In order to investigate learners' motivation, MSLQ was used. MSLQ is a self-report instrument designed to assess a general cognitive view of motivation and learning strategies [20]. According to MSLQ manual, it consists of two sections: the motivation section and the learning strategies section. Among 81 questions, 10 questions about peer learning, help seeking, and specific course were excluded because they do not fit to this study. Finally, 71 questions were used after adjusting to 5-point scale.

2.4.4.2 The ITC-Sense of Presence Inventory (ITC-SOPI)

When learners perceive presence in distance learning, learners have more sense of being in and belonging in learning [22]. To find out learners' learning experience in video-based learning environment, ITC-SOPI was used. 11 of 38 questions are not appropriate in the context of video-based learning environment, so 27 questions are only used with 5-point scale.

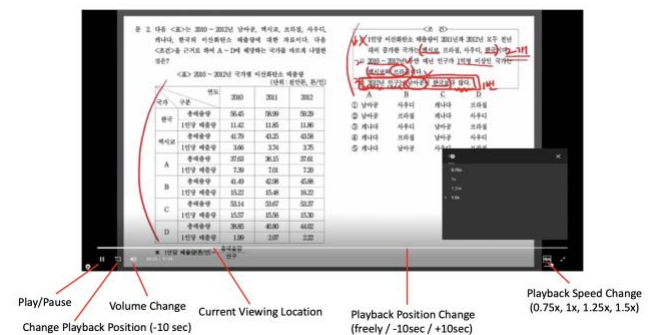


Figure 1. Example of the e-learning player

3. RESULTS

3.1 Descriptive Statistics

Table 2. Means of participants' characteristics

Variables		Mean(sd)		χ^2 (p)	t(W) (p)
		No Cue (n=36)	Cue (n=32)		
Sex (F)		17	20	.040 (.841)	
Major (STEM)		19	15	.236 (.627)	
Age		23.8 (1.80)	24.2 (1.69)		-.897 (.373)
Motivation	Intrinsic Goal Orientation	3.68 (.69)	3.59 (.79)		(.619) (.599)
	Extrinsic Goal Orientation	3.91 (.70)	3.68 (.74)		(700.5) (.125)
	Task value	4.14 (.56)	4.05 (.57)		.674 (.503)
	Control Beliefs	3.81 (.42)	3.80 (.47)		(580) (.965)
	Self-Efficacy	3.60 (.51)	3.38 (.62)		1.604 (.114)
	Test Anxiety	3.29 (.69)	3.26 (.74)		.152 (.880)
	Rehearsal	3.94 (.56)	4.00 (.60)		-.445 (.658)
Learning Strategies	Elaboration	3.98 (.52)	3.94 (.42)		(639.5) (.435)
	Organization	3.98 (.66)	3.96 (.81)		(590.5) (.863)
	Critical Thinking	3.54 (.61)	3.51 (.66)		.212 (.832)
	Metacognitive Self-regulation	3.56 (.39)	3.50 (.51)		.535 (.595)
	Time and Study Environment	3.50 (.58)	3.49 (.71)		.098 (.927)
	Effort Regulation	3.44 (.66)	3.35 (.71)		.517 (.607)

STEM: Sciences, Technology, Engineering, or Mathematics.

Total 68 participants' data were used in analysis after 9 outliers of either pupil dilation or behavioral log had been excluded. There is a difference in the participants' pre-test scores (Wilcoxon rank sum test: $U = .883$, $p < .01$). Except for this, all the other differences are not found ($p_{all} = n.s.$, see Table 2). Regarding the pre-test scores difference between two groups, the level of prior knowledge should be considered when interpreting the results.

Because the pupil dilation is affected by time goes, the data should be analyzed as time series [21]. Section division for data analysis was implemented based on four problem solving items at the 17-minute video learning. The mean and standard deviation of pupil dilation in each section were analyzed between groups (No Cue: $M_{Total} = .11(.12)$, $M_1 = .16(.12)$, $M_2 = .13(.12)$, $M_3 = .08(.12)$, $M_4 = .06(.15)$; Cue: $M_{Total} = .13(.14)$, $M_1 = .16(.16)$, $M_2 = .16(.14)$, $M_3 = .10(.15)$, $M_4 = .10(.15)$, see Table 3).

3.2 Pupillary Responses

3.2.1 Pupil dilation in time series

Average pupil dilation is gradually decreased as learning sections proceeded ($F(3, 268) = .5.834$, $p = .001$, see figure 2). The pupil size of all participants in 1st section is higher than those

Table 3. Means of dependent variables

Variables	Mean(sd)		t(W) (p)
	No Cue (n=36)	Cue (n=32)	
Total Pupil Dilation	.11 (.12)	.13 (.14)	-.656 (.51)
Pupil Dilation in 1st section	.16 (.12)	.16 (.16)	.004 (.997)
Pupil Dilation in 2nd section	.13 (.12)	.16 (.14)	-.874 (.385)
Pupil Dilation in 3rd section	.08 (.12)	.10 (.15)	(475) (.218)
Pupil Dilation in 4th section	.06 (.15)	.10 (.15)	-.873 (.386)
Behavior Frequency	18.2 (22.0)	18.8 (20.8)	(518) (.479)
Pre-test Scores	4.03 (.88)	3.47 (.95)	(760) (<.05*)
Post-test Scores	4.89 (1.04)	4.47 (1.19)	(685) (.165)
Improvement in Test Scores (= posttest - pretest)	0.86 (1.25)	1.00 (1.61)	(540) (.655)
Learning Presence	3.05 (.55)	3.16 (.52)	.813 (.419)

of participants in both 3rd and 4th sections (Tukey's post hoc: 1st vs 3rd, $p < .05$; 1st vs 4th, $p < .01$). Similarly, this tendency is shown between 2nd and 4th sections ($p < .05$). This indicates when the sections proceed, the overall pupil size is decreased.

3.2.2 The Effect of Visual Cues on Pupil dilation in time series

The result shows that pupil dilation of the group without visual cues in 1st section is higher than those of participants in both 3rd and 4th sections (Kruskal-Wallis test: $H(3) = 14.203$, $p < .01$; Nemenyi post hoc: 1st vs 3rd, $p < .05$; 1st vs 4th, $p < .05$, figure 3). This indicates that pupil dilation of the group without visual cues is statistically decreased as time goes by. However, this tendency is not shown in the group with visual cues ($F(3, 124) = 1.824$, n.s.).

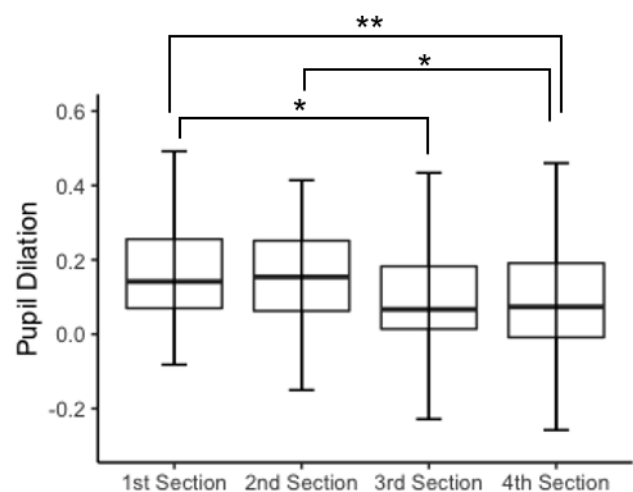


Figure 2. Pupil dilation in time series
(*, $<.05$; **, $<.01$; error bar, SEM)

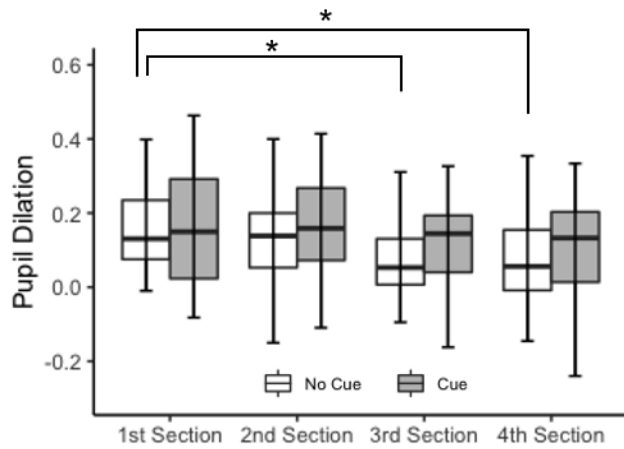


Figure 3. The effect of visual cues on pupil dilation in times series (*, <.05; error bar, SEM)

3.2.3 The Effect of Visual Cues on Pupil dilation in time series depending on self-regulation

Multiple linear regression was conducted to test the effect of visual cues on cognitive load depending on self-regulation. Results indicate that the frequencies of self-regulated behavior predict pupil dilation in the non-visual cue group in time series ($R^2 = .137$, adjusted $R^2 = .112$, $p < .001$, see Table 4). Especially, pupil dilation in 3rd section ($B = -.082$, $\beta = -.263$, $SE = .030$, $p = .007$) and 4th section ($B = -.101$, $\beta = -.324$, $SE = .030$, $p = .001$) are predicted in the non-visual cue group. Behavior frequency is also explained in the non-visual cue group ($B = .002$, $\beta = .221$, $SE = .001$, $p = .006$). On the other hand, the frequencies of self-regulated behavior do not predict pupil dilation in the visual-cue group in time series ($R^2 = .043$, adjusted $R^2 = .012$, n.s.).

4. DISCUSSION

Effective instructional design is crucial to maintain learners' motivation and promote cognitive processing in video-based learning. When an ineffective learning material is provided to

Table 4. Multiple linear regression analyses predicting cognitive load and self-regulation

Variables		B	β	SE	p
No Cue	Pupil Dilation in 1st section	.029	.000	.023	.197
	Pupil Dilation in 2nd section	-.032	-.102	.030	.290
	Pupil Dilation in 3rd section	-.082	-.263	.030	.007**
	Pupil Dilation in 4th section	-.101	-.324	.030	.001**
	Behavior Frequency	.002	.221	.001	.006**
Cue	Pupil Dilation in 1st section	.049	.000	.029	.091
	Pupil Dilation in 2nd section	-.004	-.011	.038	.921
	Pupil Dilation in 3rd section	-.060	-.170	.038	.117
	Pupil Dilation in 4th section	-.068	-.194	.038	.074
	Behavior Frequency	.000	.035	.001	.691

learners, learners have to effort to distinguish key elements from irrelevant information. Being distracted by irrelevant information causes extraneous cognitive load. Based on the CLT and SRL, cueing was used to investigate whether it moderates learners' cognitive load depending on their self-regulatory capacity.

The present study shows pupil dilation of the non-visual-cue group statistically is decreased time goes by. By contrast, there is no statistical difference in the visual-cue group as learning sections proceeded. This implies that learners' cognitive load can be affected by visual cues in time series. Visual cues have an effect on cognitive load in time series within each group. By contrast, pupil dilation was not differed between those groups. This indicates that learners in the non-visual-cue group experience difficulty to keep their motivation and put into mental effort in learning, due to ineffective learning. Next, learners' self-regulated behavior explains cognitive load in the non-visual cue group. In the visual-cue group, learners' self-regulated behavior does not predict cognitive load. Although the interaction between self-regulation and cognitive load is not figured out, the results partially suggest that self-regulation would have the effect on cognitive load within time change by showing different tendencies between two groups.

The previous research said pupil dilation is affected by time and tiredness [9], but the visual-cue group could keep deep cognitive processing and the arousal status until the end of learning. Furthermore, reduction of extraneous cognitive load and increase of germane cognitive load are expected when cue is provided [10]. Consistently, learners' germane cognitive load can be kept and increased when extraneous cognitive load is reduced by the effect of visual cues in this study. Therefore, instructional designers have to consider the effect of visual cues with time series in video-based learning.

This study has several limitations. The first limitation is the difficulty in classifying specific elements of cognitive load. Although, pupillary response is useful way to predict learners' cognitive load, pupil dilation has a problem that pupil dilation can be interpreted in two ways: an increase in germane cognitive load (or mental effort) or extraneous cognitive load (or allocation of attentional resources as task demands increased) [21]. In other words, pupil dilation can reflect both extraneous load and germane load in the same way.

The second limitation is that the group differences in pupil dilation are not figured out, though there are statistically significant differences in time series within each group. This may be caused by the difference of the level of the prior knowledge. Learners' prior knowledge is related to intrinsic cognitive load. This intrinsic load is hard to be affected by instructional design [12]. Considering this, the group's prior knowledge difference may offset the effect of the visual cues on the pupillary response.

Third limitation is about the analytic method of self-regulated behavior. Individual difference of behavior frequency was not controlled in this study. To minimize the variation, behavior frequency was analyzed by the sum over all sections, not by the division of each section. For more accurate measurements of self-regulation, the number of self-regulation behaviors should be analyzed within time goes.

Future research should continue to explore the effect of visual cues on cognitive load depending on self-regulation. With the combination of different measurements for assessing cognitive load, separating specific aspects of cognitive load is expected. When learners' cognitive load is divided and analyzed in each element of cognitive load, designing effective instruction is possible to improve learning effectiveness in video-based learning.

5. REFERENCES

- [1] Binbasaran Tuysuzoglu, B., Greene, J.A. 2015. An investigation of the role of contingent metacognitive behavior in self-regulated learning. *Metacognition and Learning*. 10, (Apr. 2015) 77-98. DOI - <https://doi.org/10.1007/s11409-014-9126-y>
- [2] Babette Park, Andreas Korbach, Roland Brünken. 2015. Do Learner Characteristics Moderate the Seductive-Details-Effect? A Cognitive-Load-Study Using Eye-Tracking. *Educational Technology & Society*. 18, 4 (Oct. 2015), 24-36.
- [3] Beau Abar, Eric Loken. 2010. Self-regulated learning and self-directed study in a pre-college sample. *Learning and Individual Differences*. 20, 1 (Feb. 2010), 25-29. DOI - <https://doi.org/10.1016/j.lindif.2009.09.002>
- [4] Christopher Lange, Jamie Costley, Seung-lock Han. 2017. The Effects of Extraneous Load on the Relationship Between Self-Regulated Effort and Germane Load Within an E-Learning Environment. *International Review of Research in Open and Distributed Learning*. 18, 5 (Aug. 2017), 64-83. DOI - <https://doi.org/10.19173/irrodl.v18i5.3028>
- [5] Conijn, R., Snijders, C., Kleingeld, A., Matzat, U. 2017. Predicting student performance from LMS data: A comparison of 17 blended courses using Moodle LMS. *IEEE Transactions on Learning Technologies*. 10, 1 (Mar. 2017), 17-29.
- [6] Ford, J. K., Smith, E. M., Weissbein, D. A., Gully, S. M., Salas, E. 1998. Relationships of Goal Orientation, Metacognitive Activity, and Practice Strategies With Learning Outcomes and Transfer. *Journal of Applied Psychology*. 83, 2, 218-233. DOI - <https://doi.org/10.1037/0021-9010.83.2.218>
- [7] Fred Paas, Juhani E. Tuovinen, Huib Tabbers, Pascal W. M. Van Gerven. 2003. Cognitive Load Measurement as a Means to Advance Cognitive Load Theory. *Educational Psychologist*. 38, 1 (Jun. 2010), 63-71. DOI - https://doi.org/10.1207/S15326985EP3801_8
- [8] Huib K. Tabbers, Rob L. Martens, Jeroen J. G. van Merriënboer. 2010. Multimedia instructions and cognitive load theory: Effects of modality and cueing. *British Journal of Educational Psychology*. 74, 1 (Dec. 2010), 71-81. DOI - <https://doi.org/10.1348/000709904322848824>
- [9] Jeroen J. G. van Merriënboer, John Sweller. 2005. Cognitive Load Theory and Complex Learning: Recent Developments and Future Directions. *Educational Psychology Review*. 17, 2 (Jun. 2005), 147-178.
- [10] Johannes Zagermann, Ulrike Pfeil, Harald Reiterer. 2016. Measuring Cognitive Load using Eye Tracking Technology in Visual Computing. *BELIV '16: Novel Evaluation Methods For Visualization*. 78-85. DOI - <https://doi.org/10.1145/2993901.2993908>
- [11] Joseph Tao-yi Wang. 2010. Pupil Dilation and Eye-tracking. *A Handbook of Process Tracing Methods for Decision Research: A Critical Review and User's Guide*. Psychology Press.
- [12] Kahneman D., Beatty J. 1966. Pupillary Diameter and Load on Memory. *Science*. 154 (Dec. 1966), 1583-1585.
- [13] Krista E. DeLeeuw, Richard E. Mayer. 2008. A Comparison of Three Measures of Cognitive Load: Evidence for Separable Measures of Intrinsic, Extraneous, and Germane Load. 100, 1, 223-234. DOI - <https://doi.org/10.1037/0022-0663.100.1.223>
- [14] Kruger, J. L., Doherty, S. 2016. Measuring cognitive load in the presence of educational video: Towards a multimodal methodology. *Australasian Journal of Educational Technology*. 32, 6, 19-31. DOI - <https://doi.org/10.14742/ajet.3084>
- [15] Krejtz K, Duchowski AT, Niedzielska A, Biele C, Krejtz I. 2018. Eye tracking cognitive load using pupil diameter and microsaccades with fixed gaze. *PLoS ONE*. 13, 9, 1-23.
- [16] Mayer, R. E. 1992. *Thinking, problem solving, cognition*. New York: NY. W. H. Freeman & Company.
- [17] Mayer, R. E. 2001. *Multimedia learning*. New York, NY: Cambridge University Press.
- [18] Michail N. Giannakos. 2013. Exploring the video-based learning research: A review of the literature. *British Journal of Educational Technology*. 4, 6 (Oct. 2013), 191-195. DOI - <https://doi.org/10.1111/bjjet.12070>
- [19] Paul R. Pintrich and Elisabeth V. De Groot. 1990. Motivational and Self-Regulated Learning Components of Classroom Academic Performance. *Journal of Educational Psychology*. 82, 1, 33-40. DOI - <https://doi.org/10.1037/0022-0663.82.1.33>
- [20] Paul R. Pintrich, David A. F. Smith, Teresa Garcia, Wilbert J. McKeachie. 1991. *A Manual for the Use of the Motivated Strategies for Learning Questionnaire (MSLQ)*. Technical Report No. 91-B-004. The University of Michigan.
- [21] Schunk, D. H., & Zimmerman, B. J. (Eds.). (1998). *Self-regulated learning: From teaching to self-reflective practice*. New York: The Guilford Press.
- [22] Tina Seufert. 2018. The interplay between self-regulation in learning and cognitive load. *Educational Research Review*. 24, 116-129. DOI - <https://doi.org/10.1016/j.edurev.2018.03.004>
- [23] Tzu-Chien Liu, Yi-Chun Lin, Fred Paas. 2013. Effects of cues and real objects on learning in a mobile device supported environment. *British Journal of Educational Technology*. 44, 3 (Jun. 2012), 386-399. DOI - <https://doi.org/10.1111/j.1467-8535.2012.01331.x>
- [24] Van Gerven, P. W. M., Paas, F., van Merriënboer, J. J. G., Schmidt, H. G. 2002. Memory load and task-evoked pupillary responses in aging. Manuscript submitted for publication.
- [25] Zargari Marandi, R., Madeleine, P., Omland, Ø. Et al. 2018. Eye movement characteristics reflected fatigue development in both young and elderly individuals. *Scientific reports*. 8, 13148, 1-10. DOI - <https://doi.org/10.1038/s41598-018-31577-1>
- [26] Itaedu. (2018, Mar 2). 2016 psat 자료해석 합격생 기출해설. YouTube. <https://youtu.be/W7iVCVB5P-c>

Towards Understanding the Impact of Real-Time AI-Powered Educational Dashboards (RAED) on Providing Guidance to Instructors

Ajay Kulkarni
George Mason University
Fairfax, VA
akulkar8@gmu.edu

Michael Eagle
George Mason University
Fairfax, VA
meagle@gmu.edu

ABSTRACT

The objectives of this ongoing research are to build Real-Time AI-Powered Educational Dashboard (RAED) as a decision support tool for instructors, and to measure its impact on them while making decisions. Current developments in AI can be combined with the educational dashboards to make them AI-Powered. Thus, AI can help in providing recommendations based on the students' performances. AI-Powered educational dashboards can also assist instructors in tracking real-time student activities. In this ongoing research, our aim is to develop the AI component as well as improve the existing design component of the RAED. Further, we will conduct experiments to study its impact on instructors, and understand how much they trust RAED to guide them while making decisions. This paper elaborates on the ongoing research and future direction.

Keywords

Decision support tool, Educational dashboard, Interactive visualizations, Impact, Unsupervised learning, Recommendations

1. INTRODUCTION

A dashboard is a collection of wisely selected visualizations that assists in understanding raw information stored in databases, which helps human cognition [6]. A dashboard can be viewed as a container of indicators [13], but Bronus et al. provided the most accurate definition of the dashboard. Bronus et al. defined the dashboard as "an easy to read, often single page, real-time user interface, showing a graphical presentation of the current status (snapshot) and historical trends of an organization key performance indicators (KPIs) to enable instantaneous and informed decisions to be made at a glance" [5]. This type of visual displays are critical in sense-making as humans are able to process large amounts of data if presented in a meaningful way [17]. The use of learning analytics tools and visualizations have the potential to provide effective support to instructors by helping them to

keep students engaged and achieve learning objectives [15]. Yoo et al. [21] conducted a review of educational dashboards in which they underline the usefulness by mentioning dashboards present the results of the educational data mining process and help teachers to monitor and understand their student's learning patterns. We can apply the same principle to the data collected from a student's quiz questions. The responses received from the quiz can be used for understanding conceptual and meta-cognitive knowledge components [4]. It has also been noted that very few of the deployed learning dashboards addressed the actual impact on instructors [20]. Thus, we see a need for a Real-Time AI-Powered Educational Dashboard (RAED) that is designed for assisting instructors. There are two main objectives of the proposed research.

Objective 1: Build a RAED, which will act as a decision support tool for instructors.

Objective 2: Measure the impact of the RAED on instructors and understand their trust in using the RAED while making decisions.

The proposed dashboard consists of two components - the visualization component and the AI component. The visualization component will present an entire classroom's actions in real-time on the dashboard. This will help instructors to answer two questions: (i) where are most of the students struggling? and (ii) on which questions are most of the students using hints? Answers to these questions can be useful for providing further explanations of certain concepts immediately after the quiz. The AI component of the dashboard will perform unsupervised learning on the collected responses. It will produce clusters of students and also generate recommendations based on the results, which will be displayed on the dashboard. These recommendations made by the AI component will also be included in the visualizations of the dashboard. These visuals will facilitate the instructors' decision-making process. For instance, an instructor may decide to give additional questions or teach a particular concept again after getting recommendations. Therefore, the current research will also be useful for understanding the usability, impact, and trust in the fusion of visualizations and AI in real-time.

Ajay Kulkarni and Michael Eagle "Towards Understanding the Impact of Real-Time AI-Powered Educational Dashboards (RAED) on Providing Guidance to Instructors" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 781 - 784

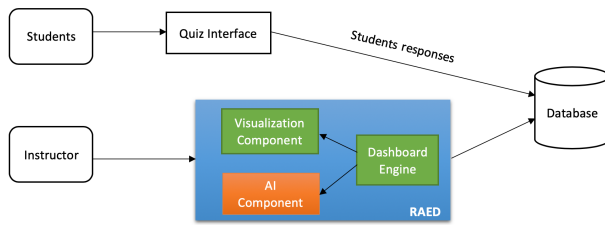


Figure 1: Proposed architecture of the RAED

2. RELATED WORK

There is a growing interest in the design and development of real-time systems, such as [14] and [19], that can provide actionable teaching analytics in real-time for decision making. These real-time systems are also beneficial from students' perspective because it gives more time to teachers to provide one-on-one support to students [11]. A user-centric teacher dashboard has been developed by Aleven et al. [2] for understanding interaction data and analytics from intelligent tutoring systems (ITS). Aleven et al. [2] noted that a dashboard could be useful to teachers for helping the class while teaching, and for preparation for the next classes. Diana et al. [7] displayed real-time analytics of interactive programming assignments on an instructor dashboard. From the results, Diana et al. [7] concluded that student outcomes could be accurately predicted from the student's program states. In addition to that, for helping more students in a classroom, Diana et al. [8] also used the machine learning model along with approach maps for identifying and grouping students who need similar help in real-time. Holstein et al. [12] developed the Luna dashboard by collecting data from interpretation sessions and affinity diagramming from middle-school teachers. The goal was to understand the dashboard's usability from the teacher's aspect, as well as its effect on students learning. In a recent paper [10], a wearable classroom orchestral tool for K-12 teachers was tested by Holstein et al. The classroom was represented as a dashboard. In that research, mixed-reality smart glasses were connected to ITSs for understanding real-time student learning and behavior within the ITSs. A framework consisting of five dimensions (Target, Attention, Social visibility, Presence over time, and Interoperation) has also been proposed for the design and analysis of teaching augmentation in [3].

3. PROPOSED CONTRIBUTIONS

This section presents the architecture and design of our proposed RAED. It further describes the features of the RAED and discusses its desirable properties, such as portability and explainability. It also includes information on the current state of our RAED development.

3.1 Architecture and design

The architecture of RAED is shown in Figure 1. Students will get a quiz interface on which they will see the questions and respond to them. The responses will get stored in a database, which can be queried by the dashboard engine. The dashboard engine will be responsible for data preprocessing and data cleaning. The resulting clean data will then be given as input to the visualization and AI components. The visualization component will produce visualizations on

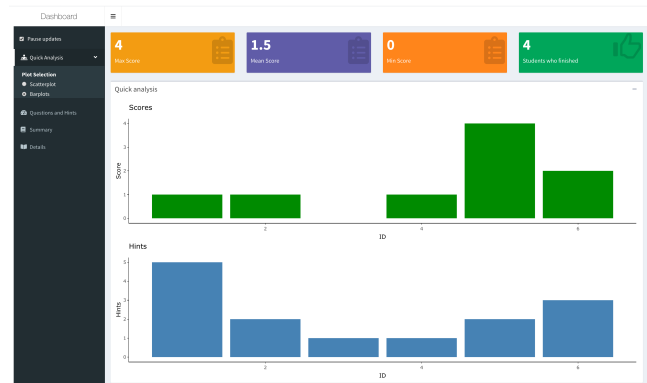


Figure 2: Design component of the RAED

the dashboard, and the AI component will perform clustering of the data in real-time. The results from the AI component will be visualized and interpreted in terms of recommendations. Currently, we have developed a quiz interface¹ for experimental purposes, using R and Shiny. This interface stores results on a Google sheet, and currently this Google sheet acts as our database. The dashboard engine is connected to the Google sheet, which queries data every 6 seconds. Thus, the dashboard is refreshed every 6 seconds.

In this on-going research, we have implemented a design component of the dashboard (shown in Figure 2), which displays real-time visualizations². The essential characteristics of the dashboards noted by Few [9] are taken into consideration while designing our dashboard. We also will be following four elements of the learning analytics process model [20] as a foundation for the conceptual design. These four elements are awareness, reflection, sensemaking, and impact. At the current state, our design includes the first three. Awareness refers to the data, which can be visualized or represented in tables as it streams. Reflection focuses on mirroring teaching practices, and sensemaking can deal with the understanding at-risk students [20].

3.2 Features of the dashboard

We propose five unique features of the RAED. We have implemented a majority of the features, and details are as follows.

1) Interactive visualizations – The visualizations generated on the dashboard are fully interactive (shown in Figure 3) and can be downloaded in Portable Network Graphics (PNG) format. Instructors can interact with them by zooming in, zooming out, selecting different components of the visualizations, etc. These visualizations can also provide meaningful information if the cursor hovers over them. Currently, our dashboard visualizations include scatter plots, bar plots, and histograms. The scatter plot and bar plots are used for understanding the quiz score and number of hints requested by students. Histograms are used to understand the score distribution of the class. Another essential role of the RAED is enhancing the perception of instructors as they can decide what to focus on.

¹<https://tinyurl.com/qnp46y9>

²<https://tinyurl.com/yx3pht5e>

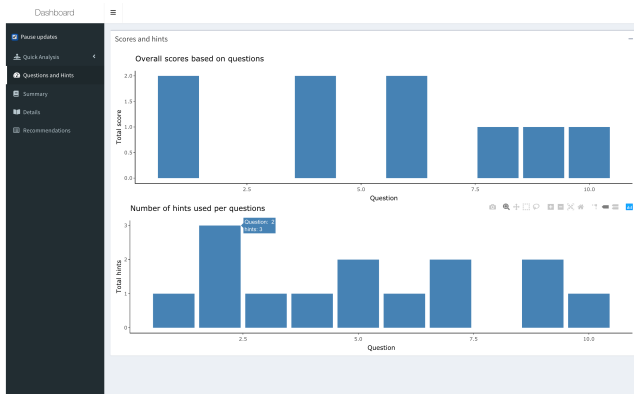


Figure 3: Interactive visualizations of the RAED

ID	Question	Response	Hint	Points
1	1	1	No	2
2	1	2	Yes	2
3	1	3	Yes	2
4	1	4	Option 1	2
5	1	5	Option 2	2
6	1	6	Option 2	2
7	1	7	Option 2	2
8	1	8	Option 1	2
9	1	9	Option 1	2
10	1	10	Option 1	2

Figure 4: Dynamic tables on the RAED

2) Dynamic tables – One of the unique features of the RAED is its dynamic tables. We have provided a summary of the class and scorecard in the form of tables. These tables get updated in real-time, and instructors can search as well as sort the tables. The RAED also provides functionality to download these tables into CSV format for further analysis (shown in Figure 4).

3) Portability – The dashboard is designed in R and deployed on the Shiny server. This dashboard is portable and can be connected to any database or tool.

4) Real-time – The dashboard provides updates of the data every 6 seconds, which can help to capture student’s real-time interaction during the quiz. Further, we provide the additional feature of pausing and resuming real-time streaming. This feature can be especially useful when analyzing the dashboard.

5) AI and explainability – This feature is currently under development. We plan to employ explainable AI on the dashboard. It will help instructors to understand how AI provides results to the dashboard, i.e., how it chooses the number of clusters and how it produces recommendations.

4. FUTURE DIRECTION

The future direction of this research is to develop a prototype of the RAED, test it in classrooms, and then conduct surveys to measure its impact and trust. Future research will be held in the following four phases.

- **Phase 1 (AI component):** Currently, we can store

student ids, names of the course topics, responses, scores, whether hints are used, and what is the total number of requested hints. We will be using this information for clustering students and generating recommendations for them. The process of clustering can be useful for focusing attention on students with similar characteristics and learning rates. This information can help instructors to form support groups within the class and to provide personalized guidance to particular students. For example, students from the high-performance group can be paired with students in the low-performance group, which can help to improve performance.

In the first step, similar students will be identified by performing clustering on the data. The goal of this step is to identify three clusters (high performance, average performance, and low performance). It is essential to visualize the process of clustering for implementing explainable AI. Thus, the implementation of Agglomerative Hierarchical Clustering makes a suitable choice. Using this approach, clustering process will begin with points as individual clusters, and at each step, the similar points will be merged into a larger cluster [18]. This entire process of clustering can be visualised by plotting a dendrogram, which fulfills our goal of explainability. The other advantage of using Agglomerative Hierarchical Clustering is that it provides good results when given small datasets as input [1], as is the expected number of students in a class. In the next step, information of students from these clusters will be obtained, and a list of concepts that students need to improve will be derived from the responses. It will also provide suggestions on pairing students during in-class activities. This information will act as recommendations to instructors and can also help to understand conceptual as well as meta-cognitive knowledge components of the class.

- **Phase 2 (Design):** We will focus on the design aspect of the RAED using the iterative design process. In this step, the prototype will be shown, and functionalities will be explained to the instructors for getting their insights on RAED. Surveys will be provided to the instructors for evaluations and to know their additional needs. Questions in the survey will be based on the questionnaire created by Park et al. [16].

The results will help us to get inputs on information usefulness, visual effectiveness, appropriateness of visual representation, user-friendliness, and understanding of the information. Changes will be made in the design after analyzing responses from the survey. In the next iteration, RAED will be shown again to the instructors, and their feedback will be requested. In this way, at the end of this phase, the prototype will be ready for testing.

- **Phase 3 (Testing):** In this phase, the prototype will be tested in classrooms. The quiz interface will be provided to students, and RAED will be made available to instructors. This phase will help us understand the technical problems that may occur, such as server issues. Improvements will be made as necessary.

- **Phase 4 (Survey):** In this final phase, surveys will be provided to instructors to understand their changes in behavior, the achievement of the objective, trust in the system, effect on motivation and decision making due to the RAED. The responses will help us to measure the impact of the RAED on the instructor's decision making. It will also give us insights about the trust instructors have in the RAED.

5. ACKNOWLEDGMENTS

The authors would like to thank DataLab at George Mason University for their support. The authors also would like to thank Dr. Olga Gkountouna for useful feedback on this work.

6. REFERENCES

- [1] O. A. Abbas. Comparisons between data clustering algorithms. *International Arab Journal of Information Technology (IAJIT)*, 5(3), 2008.
- [2] V. Aleven, F. Xhakaj, K. Holstein, and B. M. McLaren. Developing a teacher dashboard for use with intelligent tutoring systems. *technology*, 34:44, 2010.
- [3] P. An, K. Holstein, B. d'Anjou, B. Eggen, and S. Bakker. The ta framework: Designing real-time teaching augmentation for k-12 classrooms. *arXiv preprint arXiv:2001.02985*, 2020.
- [4] L. W. Anderson and L. A. Sosniak. *Bloom's taxonomy*. Univ. Chicago Press Chicago, IL, 1994.
- [5] F. Brouns, M. E. Zorrilla Pantaleón, E. E. Álvarez Saiz, P. Solana-González, Á. Cobo Ortega, E. R. Rocha Blanco, M. Collantes Viaña, C. Rodríguez Hoyos, M. De Lima Silva, C. Marta-Lazo, et al. Eco d2. 5 learning analytics requirements and metrics report. 2015.
- [6] S. K. Card, J. D. Mackinlay, and B. Shneiderman. Using vision to think. In *Readings in information visualization: using vision to think*, pages 579–581. 1999.
- [7] N. Diana, M. Eagle, J. Stamper, S. Grover, M. Bienkowski, and S. Basu. An instructor dashboard for real-time analytics in interactive programming assignments. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pages 272–279, 2017.
- [8] N. Diana, M. Eagle, J. Stamper, S. Grover, M. Bienkowski, and S. Basu. Peer tutor matching for introductory programming: Data-driven methods to enable new opportunities for help. International Society of the Learning Sciences, Inc.[ISLS]., 2018.
- [9] S. Few. *Information Dashboard Design: Displaying data for at-a-glance monitoring*, volume 81. Analytics Press Burlingame, CA, 2013.
- [10] K. Holstein, G. Hong, M. Tegene, B. M. McLaren, and V. Aleven. The classroom as a dashboard: co-designing wearable cognitive augmentation for k-12 teachers. In *Proceedings of the 8th international conference on learning Analytics and knowledge*, pages 79–88, 2018.
- [11] K. Holstein, B. M. McLaren, and V. Aleven. Intelligent tutors as teachers' aides: exploring teacher needs for real-time analytics in blended classrooms. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pages 257–266, 2017.
- [12] K. Holstein, F. Xhakaj, V. Aleven, and B. McLaren. Luna: a dashboard for teachers using intelligent tutoring systems. *Education*, 60(1):159–171, 2010.
- [13] M. Ji, C. Michel, E. Lavoué, and S. George. Ddart, a dynamic dashboard for collection, analysis and visualization of activity and reporting traces. In *European Conference on Technology Enhanced Learning*, pages 440–445. Springer, 2014.
- [14] R. Martinez-Maldonado, J. Kay, K. Yacef, M. T. Edbauer, and Y. Dimitriadis. Mtdashboard and mtdashboard: supporting analysis of teacher attention in an orchestrated multi-tabletop classroom. In *10th International Conference on Computer-Supported Collaborative Learning, CSCL 2013*, pages 320–327. International Society of the Learning Sciences, 2013.
- [15] R. Mazza and V. Dimitrova. Visualising student tracking data to support instructors in web-based distance education. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 154–161, 2004.
- [16] Y. Park and I.-H. Jo. Factors that affect the success of learning analytics dashboards. *Educational Technology Research and Development*, 67(6):1547–1571, 2019.
- [17] S. Shemwell. Futuristic decision-making. *Executive Briefing Business Value from*, 2005.
- [18] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining*. Pearson Education India, 2016.
- [19] M. Tissenbaum, C. Matuk, M. Berland, L. Lyons, F. Cocco, M. Linn, J. L. Plass, N. Hajny, A. Olsen, B. Schwendimann, et al. Real-time visualization of student activities to support classroom orchestration. Singapore: International Society of the Learning Sciences, 2016.
- [20] K. Verbert, E. Duval, J. Klerkx, S. Govaerts, and J. L. Santos. Learning analytics dashboard applications. *American Behavioral Scientist*, 57(10):1500–1509, 2013.
- [21] Y. Yoo, H. Lee, I.-H. Jo, and Y. Park. Educational dashboards for smart learning: Review of case studies. In *Emerging issues in smart learning*, pages 145–155. Springer, 2015.

Estimation for cognitive load in Video-based learning through Physiological Data and Subjective Measurement by Video Annotation

InHye Lee
Ewha Womans University
22netaurus@gmail.com

ABSTRACT

When designing a video-based learning such as MOOC, it is very important to understand the cognitive aspects of learning and reflect them in the design. Many studies use subjective and physiological data as indicators of cognitive load. To fully understand the cognitive load, we need to understand both of them simultaneously. Therefore, this study is to investigate whether eye data (Mean Pupil Dilation, Mean Fixation Duration) predicts subjective cognitive load during video learning. Furthermore, as a second research question on a broader scale, we examined whether eye data predicts high and low states of subjective cognitive load during video learning. Through this, we expected to find the possibility of Video Annotation and Eye data as a way to measure Cognitive Load during video learning. The experiment was conducted in a controlled laboratory environment with 100 students. In the video learning situation, the learner's eye data was measured using an eye tracker. Immediately afterwards, a video annotation (VA) interview technique was used to put markers according to the cognitive load types such as A (Understanding), B (Easy), C (Complicated), and D (Discomfort). The collected data will be analyzed by Support Vector Machine, a machine learning technique that is considered appropriate for the physiological data.

Keywords

Video-based learning, Physiological data, Eye data, Video Annotation, Eye tracking, Cognitive Load, Support Vector Machine

1. INTRODUCTION

Recently video-based learning has become a common form of learning for both corporate and school education as well as open contents such as MOOCs and Coursera. However, since instructors and learners are separated in time and space in video-based learning, it is difficult to immediately reflect learner's response to the instructional design. In addition, universal instructional design does not reflect the characteristics of each learner. For this reason, universal instructional design in video-based learning tends to result in learner neglect or dropout, as can be seen in MOOC's high dropout rate. Therefore, instructional design considering the learner's learning process is important in

video-based learning.

Cognitive load is the one of the most remarkable factors in human learning process. In many studies, including Moreno's work [8], we have accumulated evidence that cognitive load is a reliable factor for effective video-based learning design. In addition, various data left by learners during video-based learning are important resources for instructional design considering individual cognitive load. However, most pedagogical studies measure the learner's internal processes using a psychometric scale. This approach has problem to be solved that memory distortion may occur because it is usually measured after learning. Hence, physiological data are used as an objective measurement index. Especially, eye data can be measured in real time. Moreover, in case of pupillary reflex, it is under control of autonomous nervous system and cannot be voluntarily controlled by the subject. However, despite its advantages, it is sensitive to environmental variations such as luminance [3][5][18]. Therefore, using both psychometric subjective scales and eye data can complement each other.

In this study, we will examine how the physiological data predicts the subjective measurements of learners using Support Vector Machine which is machine learning techniques. Also, in case of subjective measurement, video annotation is used to prevent memory loss after learning. This study proceeded with Video Annotation right after eye tracking experiment. This study aims to discover the possibility of using both of physiological data and Video Annotation to measure cognitive load reliably. Through this study, we expect that indicators of cognitive load will be used as more reliable resources for video-based instructional design.

2. LITERATURE REVIEW

2.1 Cognitive Load Theory

According to the Cognitive Load Theory proposed by Sweller [9], Cognitive Load is defined as the sum of mental activities imposed on human working memory when processing new information. There are three types of cognitive loads, and each type of cognitive load is additive. First, external cognitive load is a cognitive load imposed by inappropriate instructional design, which can be controlled by efficiently structured and designed learning environment and tasks [11].

Since external cognitive load is not a desirable load for learning, it needs to be minimized through instructional design. Second, the intrinsic cognitive load is the cognitive load caused by the element interactivity of the learning contents. As the task becomes more complicated, the intrinsic cognitive load felt by the learner increases. Finally, the germane load is the cognitive load used to handle schema acquisition and automation in learning. Germane

In-Hye Lee "Estimation for cognitive load in Video-based learning through Physiological Data and Subjective Measurement by Video Annotation" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 785 - 789

load is a very important load to facilitate learning. If these three types of cognitive loads exceed working memory, information processing, including learning, will be threatened. Therefore, the purpose of instructional design is to minimize the external cognitive load and increase the intrinsic load within the learner's memory capacity. Instructional design considering each type of cognitive load presented by Cognitive Load Theory can provide more customized learning for individual students[19].

Cognitive Load Measurement Methods

The development of a method to measure cognitive load more effectively plays an important role in the study of cognitive loads [10]. Researches measuring cognitive load are largely divided into subjective and objective measurement methods. First, the subjective measurement method is measured through rating scales. This subjective item tool has the advantage of being able to measure each element of cognitive load separately. It is also widely used to measure cognitive load in many studies because it can be measured by learners relatively easily[13][14]. Cognitive Load mainly uses subject ratings of mental effort and task difficulty as indicators of cognitive load[12]. However, the subjective scale has a disadvantage in that the memory is distorted because it is influenced by the learner's bias and occurs after learning. In addition, there are objective measurement methods measuring through physiological data using EEG, eye data and skin sensitivity. In particular, the eye data measured by the eye tracker has attracted attention in recent studies[18]. Since eye data is measured at the moment of learning, it can be measured without affecting learning process. However it has a problem to be solved that it is sensitive to the external environment.

The preceding studies comparing subjective data and eye data are as follows. Korbach, Brünken and Park[6] set up three groups that distinguish external, intrinsic, and intrinsic cognitive loads, and then identified the differences through rhythm methods, subjective ratings, and eye data. As a result, both objective and subjective measures significantly distinguished the differences between groups. Most of the studies that classify cognitive load use subjective measurement methods or use both objective and subjective measurements together. However, studies that deal with both subjective and objective data should be preceded more by classifying cognitive load types in various learning contexts.

2.2 Eye data

This study utilizes Pupillary data and Fixation Duration among eye data. For Pupil data, Mean Pupil Dilation(MPD) is used. MPD has been used as a reliable indicator of cognitive load. In most cases, MPD expands in according to increasing cognitive load[1]. However, as mentioned earlier, it needs to be careful when collecting because it responds to not only psychological changes but also visual stimuli caused by environmental changes. Marquart & Winter[7] measured cognitive loads while the driver was driving using blinking eyes, eye fixation, and pupil dilation indicators. As a result, the expansion of the pupil size was observed statistically when the workload occurred during operation. Fixation Duration also can be used to measure the attention that individuals have paid to stimuli which means that Fixation Duration can be one of factors that increase cognitive load[16].

2.3 Video Annotation

Video Annotation is a kind of retrospective technique. Retrospective technique is a follow-up observation method that uses visual or auditory clues to access a subject's memory and recall the thoughts and strategies that occurred while performing a specific action or task[2]. The video annotation presents the learning video to the learner immediately after the learning ends for recall. Learners stop the video where they want it and record their thoughts[4]. It is a promising approach to facilitating video retrieval but also it can avoid the intensive labor cost of pure manual annotation[17].

The study used Techsmith's Morae software for video annotation. Notes settings were A(Understanding) which indicates germane load, B(Easy) and C(Complicated) which indicate low and high intrinsic load and D(Discomfort) for extraneous cognitive load.

3. METHODS

3.1 Research Question

The research questions of this study are as follows.

1. Does eye data (pupil response, eye fixation duration) predict subjective cognitive load during video-based learning?
2. Does eye data (pupil response, eye fixation duration) predict the total amount of subjective cognitive load(high and low) during video-based learning?

3.2 Experiment Settings

This study was approved according to the Institutional Review Board (IRB) Institutional Review Board (EBH), and was conducted on 100 male and female college students.

3.2.1 Procedure

This study was conducted in the Edutech Convergence laboratory at Ewha Womans University in order to provide an optimal environment in consideration of illuminance and noise that affect measurement data. The window in the laboratory was covered so that the laboratory was not affected by light intensity. The height of the chair and the pedestal were adjusted to each subject just before measurement, so that the environment of the participant's pupils was accurately tracked. The experiment time for each subject lasted about 100 minutes and one person at the same time.

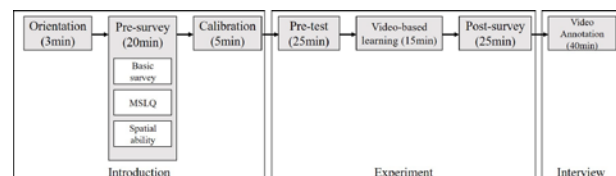


Figure 1. Experiment procedure

As shown in Figure 1, the study purpose, the duration of the experiment, and the precautions related to the experiment were announced before entering the controlled experiment site. Subsequently, the basic personal information and learning motivation strategy which is called MSLQ[15]. They were collected through the 5 Likert questionnaire. After the survey, the subject enters a laboratory where light and noise are controlled. The controlled laboratory is shown in Figure 2.



Figure 2. Environment setting for eye tracking



Figure 3. Environment setting for video annotation

According to Tobii eye tracker's manual, it was recommended to keep the distance between the study subject and the measuring device constant for accurate measurement. Chin pedestal was used to maintain constant distance. Concerned about an increase in the subject's fatigue, they were guided to rest for 30 seconds at the end of each step. In each step (pretest, video study, posttest), the sequence was set to stare at the front for 10 seconds to find the baseline of the pupil.

As shown in Figure 3, video annotation was performed in the room prepared for the interview. To implement a special interview method called video annotation, we used TechSmith's Morae program.

3.2.2 Participants

This study estimates the cognitive load in the video-based learning situation and recruits the most accessible adults (college students) in the video-based learning context such as MOOCs and Coursera. For accurate measurement, only those who do not have eye-related diseases and who can replace glasses when wearing lenses were allowed to participate in the experiment. In addition, the gender and major categories were selected to be evenly distributed. Even if there were no eye-related diseases, if the eye tracker did not track the eye during calibration, the ear was taken because no data could be collected. In addition, subjects whose data exceeded the recommended range for eye tracking during pretreatment or missing more than 50% of the data due to missing values based on pupil range outliers were excluded from the analysis.

Of the total 100 participants, 96 participated endlessly without returning home halfway. Among them, 82 were studied except for missing values. Thus, 18% of the participants were excluded from the analysis. The demographic information of the study subjects used for analysis in this study is as follows.

Table 1. Demographic Information of Subjects(n=82)

Gender	Male	39
	Female	43
Major	Liberal arts	42
	Science and Engineering	40

3.2.3 Stimuli

The stimuli given to the subject during eye tracking are as follows.

Pre-test and post-test

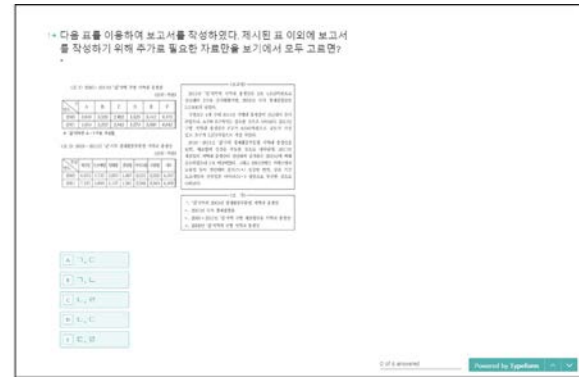


Figure 3. A screen shot of pre-test

The pre- and post-examination tests consisted of PSAT questions, a test to select South Korean civil servants. The problem is that both the pre- and post-test have a total of six questions and the time limit is 25 minutes.

Video-based learning

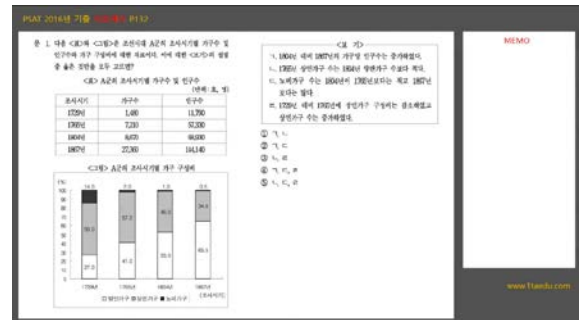


Figure 4. An example of video screen

As shown in Figure 5, video-based learning is a learning video that teaches PSAT problem solving strategies. In order not to distract the learners' attention other than the contents of the study, the lecturer was selected as a video in which the lecture was conducted only by voice.

3.2.4 Instruments for measurement

Eye tracker

In this study, Tobii Pro X2-30 (30Hz) eye tracker was used to measure 30 frames per second to measure pupil response and eye fixation duration. As can be seen in Figure 2, pupils can be measured non-intrusively simply by keeping the 50 ~ 70cm distance between the measuring device and the subject without additional wearing. Therefore, data can be collected without pre- and post-testing and video-based learning. The collected data was extracted in the form of csv data that can be analyzed using Tobii Pro Studio program, Excel and R Studio program.

Morae for video annotation

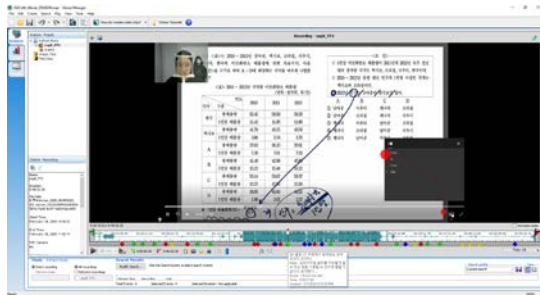


Figure 5. A screenshot of Morea for video annotation

Figure 6 shows an example of TechSmith's Morae program. As you can see from the enlarged figure, Morae program screen is replayed by the learner at the time. In the upper left corner, the face image of the learner was taken in the previous experiment. The screen shows the points and paths that the subject looked at then with red dots and lines..

The learner entered the thoughts and feelings he had heard at the moment while looking at the screen, his gaze, and facial expressions that he stared at in the previous experiment. Press Ctrl + M to select one of A (Understanding), B (Easy), C (Complex), or D (Discomfort), and then type the sentence directly for the reason. During the direct entry by the subject, the researchers looked at the screen together and helped when confused about which markers to choose.

3.3 Data Analysis

3.3.1 Datasets

The variables selected for the research question are shown in Table 2 below.

Table 2. Variables(Eye data and video annotation)

Variables		Description
Eye data	LocalTime Stamp	Timestamp counted from the start of the recording
	GazeEvent tType	Type of eye movement event classified by the fixation filter settings
	GazeEvent Duration	Duration of an eye movement event
	Distance Right/Left	Distance between eyes and the eye tracker
	Pupil Right/Left	Estimated size of the right(left) eye pupil
	Validity Right/Left	Indicates the confidence level that eyes have been correctly identified
Video Annot ation	Elapsed Time	Timestamp when subjected noted markers
	Details	Indicate markers which subjects noted

3.3.2 Data pre-processing

As mentioned earlier, even if the experiment was completed, the subjects whose data were more than 50% that could not be used for analysis due to missing values in the preprocessing process were excluded from the analysis. Eye data was excluded from the analysis when the recommended distance was out of 50 ~ 70cm. In the case of the pupil size, the difference between the pupil size

of the left and the right is more than 0.4 mm and was also determined as the pupil portion.

In the case of the pupil size, the average of the baseline measured in each section was calculated, and then derived by subtracting the baseline from the measured pupil size. Therefore, because the pupil size measured every second is the baseline minus, the pupil expansion indicators are positive when the pupil is larger than the baseline, but may be negative when the pupil is reduced.

After that, the LocalTimeStamp and Elapsed Time variables of the eye data were changed to the same time expression, and then preprocessed by matching the note and eye data shown by time zone. Therefore, the variables used for the actual predictive analysis are as follows: Markers from Video Annotation (VA), Mean Pupil Dilation (MPD), Mean Fixation Duration (MFD)

The first research question analyzed SVM classification of preprocessed data. The second study divided the A (Understanding), C (Complicated), and D (Discomfort) groups with high cognitive loads and the B (Easy) markers with low cognitive loads. Since the high group of the three markers combined had more than three times the number of data, we randomized and set the same number as the low group. After that, we checked whether the high and low groups were predicted.

3.3.3 Data Analysis

The analysis was performed using R Studio, a statistical analysis program. The preprocessed data is analyzed by Support Vector Machine (SVM) technique. The reason for analysis by SVM method is as follows. First, because of experimental data characteristics. Since eye data is measured at 30 frames per second for 82 subjects, a very large amount of data is collected. In addition, due to the nature of the physiological data, even a laboratory set up is susceptible to microenvironmental influences. Therefore, we chose SVM that is less affected by outliers and has higher accuracy. Second, SVM has the advantage of less overfitting than other neural network techniques. Finally, it is suitable for the markers (A, B, C, D) variables obtained through video annotation because they can be classified and predicted simultaneously.

The first research question analyzed SVM classification of preprocessed data. The second study divided two groups with high cognitive loads which consists of A (Understanding), C (Complicated), and D (Discomfort) and low cognitive loads which consists of B (Easy) markers with low cognitive loads. Since the high group of the three markers combined had more than three times the number of data, we randomized and set the same number as the low group. After that, we checked whether the high and low groups were predicted.

5. REFERENCES

- [1] Beatty, J., & Lucero-Wagoner, B. (2000). The pupillary system. *Handbook of Psychophysiology*, 2, 142-162.
- [2] Colasante, M., & Douglas, K. (2016). Prepare-Participate-Connect: Active learning with video annotation. *Australasian Journal of Educational Technology*, 32(4). DOI=<https://doi.org/10.14742/ajet.2123>
- [3] Di Stasi LL, Antolí A, Cañas JJ. 2011. Main sequence: an index for detecting mental workload variation in complex component analysis study. *International Journal of Psychophysiology*, 77(1), 1-7. DOI=<https://doi.org/10.1016/j.ijpsycho.2010.03.008>

- [4] Ethel, R. G., & McMeniman, M. M. 2000. Unlocking the knowledge in action of an expert practitioner. *Journal of Teacher Education*, 51(2), 87-101. DOI=<https://doi.org/10.1177/002248710005100203>
- [5] Jainta, S., & Baccino, T. 2010. Analyzing the pupil response due to increased cognitive demand: An independent tasks. *Appl Ergonomics* 42. 807–813. DOI=<https://doi.org/10.1016/j.apergo.2011.01.003>
- [6] Korbach, A., Brünken, R., & Park, B. 2018. Differentiating different types of cognitive load: A comparison of different measures. *Educational Psychology Review*, 30(2), 503-529.
- [7] Marquart, G., Cabrall, C., & de Winter, J. 2015. Review of eye-related measures of drivers' mental workload. *Procedia Manufacturing*, 3, 2854-2861.
- [8] Moreno, R. 2010. *Cognitive load theory: Historical development and relation to other theories*. In J. L. Plass, R. Moreno, & R. Brünken (Eds.), *Cognitive load theory*, 9–28. Cambridge University Press. DOI=<https://doi.org/10.1017/CBO9780511844744.003>
- [9] Sweller, J. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257-285.
- [10] Sweller, J. 2018. Measuring cognitive load. *Perspectives on medical education*. 7(1). 1-2. DOI=<https://doi.org/10.1007/s40037-017-0395-4>
- [11] Sweller, J., Ayres, P., & Kalyuga, S. 2011. *Cognitive Load Theory*. New York, NY, US: Springer.
- [12] Paas, F. G. 1992. Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. *Journal of Educational Psychology*, 84(4), 429. DOI=<https://doi.org/10.1037/0022-0663.84.4.429>
- [13] Paas, F. G., Van Merriënboer, J. J., & Adam, J. J. 1994. Measurement of cognitive load in instructional research. *Perceptual and Motor Skills*, 79(1), 419-430.
- [14] Paas, F., Tuovinen, J. E., Tabbers, H., & Van Gerven, P. 2010. Cognitive load measurement as a means to advance cognitive load theory. *Educational Psychologist*, 38(1), 63-71. DOI=https://doi.org/10.1207/S15326985EP3801_8
- [15] Paul R. Pintrich, Davide A. F. Smith, Teresa Garcia, and Wilbert J. McKeachie. 1991. *A Manual for the Use of the Motivated Strategies for Learning Questionnaire (MSLQ)*
- [16] Vertegaal, R., & Ding, Y. (2002, November). Explaining effects of eye gaze on mediated group conversations: amount or synchronization?. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*. 41-48.
- [17] Wang, Q., Yang, S., Liu, M., Cao, Z., & Ma, Q. 2014. An eye-tracking study of website complexity from cognitive load perspective. *Decision support systems*, 62, 1-10.
- [18] Winn, M. B., Wendt, D., Koelewijn, T., & Kuchinsky, S. E. 2018. Best practices and advice for using pupillometry to measure listening effort: An introduction for those who want to get started. *Trends in hearing*, 22. 1-32. DOI=<https://doi.org/10.1177/2331216518800869>
- [19] Young, J. Q., Van Merriënboer, J., Durning, S., & Ten Cate, O. 2014. Cognitive load theory: implications for medical education: AMEE Guide No. 86. *Medical teacher*, 36(5), 371-384.

Exploration Maps, Beyond Top Scores: Designing Formative Feedback for Open-Ended Problems

Aditi Mallavarapu
University of Illinois at Chicago
Chicago, IL
amalla5@uic.edu

Leilah Lyons
New York Hall of Science
Corona, NY
llyons@nysci.org

ABSTRACT

Learners are being exposed to abstract skills like innovation, creativity and reasoning through collaborative open-ended problems. Most of these problems, like their real-world counterparts, have no definite starting or ending point, and have no fixed strategies to solve them. To help the learners explore the multiple perspectives of the problem solutions there is an urgent need for designing formative feedback in these environments. Unfortunately, there are barriers to using existing EDM approaches to provide formative feedback to learners in these environments: (1) due to the vast solution space, and the lack of verifiability of the solutions it is impossible to create task and expert models, thus making the detection of the learners progress impractical; (2) formative feedback based on individual learner models does not scale well when many learners are collaborating to solve the same problem. In this work, we redefine formative feedback as reshaping the learning environment and learners' exploration paths by exposing/enlisting "fugues" as defined by Reitman [28]. Through a case study approach we, (1) validate methods to extract learners' "fugues" from a collaborative open-ended museum exhibit, (2) design formative feedback for learners and educators using these extracted fugues in real-time, (3) evaluate the impact of exposing fugues to group of learners interacting with the exhibit.

Keywords

Data-driven, Formative feedback, Open-ended learning environments, Ill-defined problems, collaborative learning

1. INTRODUCTION

With the recent advances in storage and retrieval methods of data and increased computing power the way we look into learning processes has changed [2]. We are now able to collect data to the most minute detail which was not possible in the absence of tracking devices and computer-based interactive learning environments. These improvements in instrumentation make rich interactive "classrooms

of the future" [32] amenable to computer-driven monitoring and support, but it's not a matter of just applying existing analytic approaches. The vast majority of learning analytic techniques are predicated on assumptions about learning environments that may not hold. While there have been many examples of using data mining to track students' progress through interactive learning environments using log files (e.g., [12, 1, 23, 10, 27, 3, 5, 14, 6, 24]) most of these learning experiences have been developed with reference to expert envisioned solutions which act as a strong model that the learners need to follow [9]. For example, learners are given a well-defined, fixed goal with known, optimal number of steps to reach this goal, and known, fixed number of choices that can be made by the learner at each step. In such circumstances any user action can easily be judged as taking them closer to or farther away from the goal [20]. This clarity often underpins the structure of model based Intelligent Tutoring Systems (ITS), which typically combine exhaustive, a priori "strong" models of the content domain and prior learner performances with models of the student's current progress to generate guidance [35]. These well-constrained problem spaces have successfully been used by data miners, who rely on a priori models and on post hoc analysis to provide formative feedback to the students [10, 3] or to their teachers [23], to provide formative feedback to the environment designers [12, 23], or to provide evaluative feedback on the nature and scope of mistakes made by learners in the environment [1, 27].

However, these constrained problem spaces often do not reflect problems found in the real world. Real-world problems often possess multiple solutions, where each can be attempted with multiple alternative theories, or sometimes lack the theories to verify solutions; or possess multiple task structures leading to overlapping sub-problems which thus demand novelty rather than replication from the learner [19, 9]. Additionally, these problems are often solved by groups of people who each bring in a new perspective, perspectives which are critical to preserve in order to develop workable solutions for real-world problems like climate change, social change and many others. Due to the varying dimensions wherein open-endedness can exist, the notion of open-endedness is quite vague, and oftentimes it is difficult to differentiate open-ended problems from well-defined problems; in actuality these problems seem to exist on a continuum. Simon [31], defined open-ended problems as having three features: (1) indefinite starting points, (2) indefinite ending points, which constitute goals - either are not clearly de-

Aditi Mallavarapu and Leilah Lyons "Exploration Maps, Beyond Top Scores: Designing Formative Feedback for Open-Ended Problems" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 790 - 795

finer or are complex and imprecise, and (3) with no clear strategies to solve the problem. While presenting learners with simplified and constrained problems can be a good way to help them come to understand the core properties of a domain, exposing learners to less constrained, more open-ended problems can help them get experience with disciplinary processes and dispositions [20]. Owing to the recommendations of many educational standards both formal and informal educational settings are giving the learners opportunities to practice these disciplinary processes and develop disciplinary dispositions by exposing the learners to open-ended, project-based student centered approaches [29]. Since these problems expose the learners to a rather large solution space, some researchers have argued that open-ended learning environments, and the exploratory learning styles often promoted to go along with them, are simply not workable in educational settings [17]. While other educational researchers have argued that rather than giving up on exposing learners to open-ended problems, educators and researchers should instead seek to support learners in their explorations via proper supports [13], like scaffolds and formative feedback. We argue that data mining offers great potential for supporting open-ended learning via data-driven formative feedback.

1.1 Formative Feedback

Formative feedback has been defined as “information communicated to the learner that is intended to modify the learner’s thinking or behavior for the purpose of improving learning” [30]. Formative feedback has been used extensively to support learners while solving well-defined problems, as when a learner is given a hint based on his “distance” from the goal, or a suggestion based on the expert model to get him “closer” to this model. These forms of formative feedback are inherently tied to an assumption of one fixed goal, making them unsuitable for open-ended problems which can have a dynamically evolving state space thus demanding that the learners’ goals evolve with them. Moreover, it can be challenging to fit a fixed goal perspective to collaborative learning environments, both pragmatically (instrumentation is a challenge) and conceptually (how one can go about ascribing “credit” to multiple learners when they jointly create a solution - is a theoretically undefined proposition - we don’t yet have theories of learning, and thus metrics, that fully account for and embrace the multifaceted ways groups of learners support one another and their joint endeavors).

Summarizing, there are two main barriers to adapting existing formative feedback approaches for use in open-ended collaborative learning environments: (1) due to the vast solution space, and the frequent lack of solution verifiability, it is difficult to create an exhaustive task or expert models, thus making the detection of the learners’ progress challenging [25]; (2) Most of the formative feedback in well-constrained problems is based on individual learner models, which does not scale well when many learners are collaborating to solve the same problem, a known problem in the field. To supply formative feedback for open-ended collaborative learning environments, a fundamental re-conceptualization of how formative feedback is structured, and the techniques used to distill it from collected data, are needed.

2. PROPOSED CONTRIBUTION

This research proposes to re-conceptualize formative feedback (and how it is derived from logged data) so as to support learners in open-ended, collaborative learning environments. First, we make the deliberate decision to step away from measuring or modeling individual learner “progress” - rather than placing the learner, and his or her actions, at the center of our analytics, we instead place the *solution space* at the heart of our analysis. We conceptually relate formative feedback for open-ended problems to what Lynch et al. [19] called *Discovery Support Systems*. We redefine it to be about modeling/capturing the aspects of the problem space explored by the learner(s) so far - and how the course of that exploration has unfolded so as to have exposed aspects of the problem space to the learner.

There has been considerable research on design and impact of feedback in both ill-defined tasks and problems (E.g [4, 8, 11, 26, 9]) with methods like partial task models and in well-defined ([16, 7]) tasks and problems with model-based, constraint-based and expert solution-based approaches (See Lynch et al. [19], Fournier-Viger et al. [9] for extensive review). However, these studies have designed feedback by constraining some aspects of the open-ended problem making it less open-ended [9] and have reported findings and issues related to feedback design as very complex and often mixed [19]).

The purpose of our proposed data-driven formative feedback methods is to empower the learners themselves to reshape their exploratory path through the problem space such that it can be made more amenable for exposing learners to critical events, insights, and contrasts. These can range from simple evaluative feedback suggesting “correct” or “incorrect” where verifiable solutions are available, to complex elaborate maps illuminating the trajectory of exploration, or even tying the highlights of the exploration path with external concepts and theories. As an analogy - if prior methods of formative feedback are akin to giving a tourist step-by-step directions to reach a destination, we are attempting to produce an annotated map. We thus re-situate the problem-solving decision-making with the learners themselves, and see our mission as providing them with relevant, situationally-salient information to make those decisions.

We desire to give learners a sense of how their explorations map to the larger space of possibilities within the learning environment. In a truly open-ended learning environment, the space of possible action may be infinite, but there are often nonetheless common repeating patterns in action-response pairings. The more data we collect on how learners make use of a given learning environment, the better our map of the problem space - much like a travel guide that has been annotated by multiple tourists. To conceptualize what it means to provide a data-derived “annotated map” to learners in open-ended environments, we thus lean on the “fugue” construct developed by Reitman [28]. In music, a “fugue” is a short melody or phrase which is taken up by other instruments. We argue that data mining can be used to detect “fugues” developed by prior learners in response to certain situations within the problem space, “fugues” that could be presented to new learners as potential directions to pursue. Additionally, the “fugue” concept can be used to help learners reflect on their own exploration paths: are

they relying on very similar “melodies”, or branching out and trying new compositions? The value of the “fugue” concept is that it is not in contradiction with a multi-learner environment - the piece of music, as produced by the whole orchestra, is the subject of analysis.

2.1 Research Questions

We follow Reitman [28] recommendation of conceptualizing the problem solving in open-ended learning environments as “fugues” (like in music) where the learners could adopt a component solution and successively develop interweaving parts to that component of the solution. This leads us to:

RQ 1: What kinds of methods can be used to design domain-independent formative feedback to enable exploration and conceptualization of such “fugues”?

The idea of adopting problem solving in this manner implicitly includes metacognitive support (by providing learners with an exploration map of known “fugues”), and implicitly invites collaboration where the developmental work of one group can be picked up and further developed by others. We explore what features of the problem-solving would best motivate this kind of learning and how we could exploit these features through data-driven methods to design formative feedback.

RQ 2: How do these “fugues” of solutions repositories evolve as we expose more solutions? What limitations and advantages evolve as we expose more solutions?

As more and more groups attempt the problems the repository of known “fugues” expands, thus surfacing new possibilities for formative feedback and teaching methods in the existing domain like (1) a view into how learning, collaboration, innovation takes place in such ill-defined domain, (2) provide guidance for intervention by any humans-in-the-loop (e.g., educators), (3) use the repository to design context-specific (for learners directly) feedback for known actions/tasks, and motivate the design for similar domains by laying down foundations for (4) for the design of Intelligent Tutoring Systems which might not have a expert model readily available, and (5) for designing adaptive learning environments for ill-defined problems, where the problem can change difficulty level by tracking learners to have explored certain paths or length of paths. For my dissertation, I will explore the utility of formative feedback for the in-domain applications ((1)-(3)). However, the expansion of the “fugues” also references potential limitations of the methods, for example (1) the running time constraints for processing the data- a real-time formative feedback poses certain limitations on the time spent in processing the result which makes effectiveness rather than efficiency of the feedback a priority, (2) detecting and referencing the most commonly occurring “fugue” from/to the learners might potentially indicate tunnel vision, so helping the learners diversify “fugues” while preventing recursion problem must take precedence. Maintaining an effective balance to resolve these limitations would also be the scope of my dissertation.

RQ 3: What impact does the use of these “fugues” based formative feedback have on the learning opportunities in an open-ended learning environment?

We would like to measure the impact of the redefined formative feedback for open-ended learning environments designed in RQ 2 (with visitors and the educational staff) to validate our conceptualization and usefulness of formative “fugues” in aiding exploration and the effects of scaling on the formative feedback.

2.2 Case: Collaborative Open-Ended Simulation Based Museum Exhibit

We propose to design formative feedback for a mixed reality, simulation-based museum exhibit. Connected Worlds is an open-ended complex systems exhibit that can support up to 50 simultaneous users to explore and manipulate the ecosystem. Visitors interact with the simulation by diverting the flow of simulated water on the gallery floor, and by planting seeds in the biomes simulated on the wall projections. They are tasked with maintaining the diversity of four different biomes via planting and managing water resources. It serves as a good testing ground because the exhibit does not provide the learners with fixed goals or constraints for strategies encapsulating the two characteristics of open-ended learning environments: no verifiable solutions or end goals and no clear strategies to solve/ maintain the diversity. The visitors have to constantly work together to maintain the diversity and manage resources in the ecosystem, and there can be a varied different ways of doing the same, with interaction of the actions varying substantially across contexts nominally of the same type, producing different results across-context, a recognized quality of an open-ended task [33].

2.3 Preliminary Work and Future Directions

We have designed and built a system for unobtrusively collecting the “collective” interaction data while the visitors groups interact with the system and with each other, which is undergoing iterations to capture more facets of data. In prior work, Mallavarapu et al. [21] the data capturing system was validated by the use of a mobile interface to visualize the data for visitors, and the study showed that formative feedback influenced the problem-solving strategies the visitors were using. This study helped us establish the impact of formative feedback in an open-ended learning environment like our test site. In addition to the experimental and control contrasts in the above study we have collected interaction log-data from 32 school sessions which can be used for post-hoc processing, identification, validation of methods to design formative feedback. We propose a taxonomy of methods that can be used on the well-defined to ill-defined continuum to design formative feedback (See Table 1). Another work has recently used the school groups data to study and decipher the temporal cause-effect relationships between the learners’ collective interactions and the systemic responses [22]. This provided a conceptualization to the design of *Prediction based Feedback* (See Table 1). Our immediate future efforts will focus on applying and validating these methods with the current data in extracting and designing formative feedback for this environment. These methods will then be used through the mobile device to evaluate the impact on the exploration taking place in the exhibit.

3. ADVICE SOUGHT

1. What validation methods can be used to evaluate the methods that can extract the “fugues”? We acknowl-

Type of formative feedback	Information needed by recipient	Information needed by Analytics	Applicable analytic approaches	Applicable to "fugues"
Model based Feedback	Next steps to take, demonstration of a certain step, evaluation of skills and actions, Progress towards goals	"Goal" decomposition tree, Correct example(s), metric of correctness, task to action and skill mapping	Knowledge tracing map from goals to skills and tasks, expert models.	
Violation based Feedback	"Favourable" actions and "distance" from goals due to the action	Set of constraints on the "correct" behavior, Rules for task	Detecting when certain rule is violated.	
Sequence based Feedback (showing only ongoing interaction)	Solution paths, actions on the path	Definition of what constitutes a solution path, temporal order of <i>current</i> actions	Sequence mining where consequences of frequent previously seen sequences can be used as feedback.	X
Prediction based Feedback (showing only ongoing interaction)	Predicted Consequence of actions	Causal model of the learner interactions	Causal Inference, Regression.	X
Contrast based Feedback	examples that contrast on one or more dimensions of goals	Definition of dimension(s) of contrast and highlights for goals	Sequence mining, Ability to extract "Highlights", goals from interactions, clustering using defined dimensions of contrasts.	X
Trajectory based Feedback	A exploration map of path travelled placing them on continuum of paths	definition of dimension(s) of path characterizations, differentiating metrics	Sequence mining, Ability to extract/ differentiate trajectories, clustering.	X
Task/Events based Feedback	attempted tasks/ uncovered events on the trajectory	definitions of tasks and/ or events and their temporal order, definitions of trajectory	Ability to extract "tasks" from the actions, constituting them as trajectories, clustering depending on the definitions of tasks.	X
Comparison based Feedback	collection of trajectories attempted (till now)	definitions of trajectories, differentiating metrics	Ability to extract/ differentiate trajectories, clustering.	X

Table 1: Types of Formative feedback, information conveyed by them and details of the methods for the continuum of Learning Environments

edge that we are using the existing EDM methods to validate their applicability to our problem-space. We would want to evaluate each method for the same.

2. What other external factors need consideration when designing formative feedback for open-ended learning environments. For example, when designing formative feedback for problems tackled by individual learners researchers have explored the effect of individual differences [15, 34] on the impact of formative feedback through individual learner models; While evaluating the impact of formative feedback in collaborative open-ended learning environments - what factors do we need to consider?
3. Should we consider to establish a generalizability to this redefinition of formative feedback and its impact by validating our approach through another equivalent environment?

4. CONCLUSIONS

As we move from individually tackled well-defined problems to open-ended real-world problems to allow the 21st century learners to explore with their peers, we also need to make a move from "solution" based formative feedback to a more *Socratic* method [18] of providing feedback to enable explorations, thus giving the learners an opportunity to contemplate the implications of their decisions. Working synergistically with the learning environment the feedback should expose to the learners the opportunities to learn and practice abstract skills like reasoning, creativity, innovation, and encouraging the same in a collaborative environment. We identify the feedback for collaborative open-ended learning environments to have characteristics like meta-cognitive support, support for "collective" learner efforts, and be predicated on the characteristics of the problem space rather than the learner actions.

5. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Nos. 1623094 and 1822864,

and would not have been possible without the aid given by museum staff.

References

- [1] E. Andersen, Y.-E. Liu, E. Apter, F. Boucher-Genesse, and Z. Popović. Gameplay analysis through state projection. *Proceedings of the Fifth International Conference on the Foundations of Digital Games - FDG '10*, pages 1–8, 2010.
- [2] B. Bakhshinategh, O. R. Zaiane, S. El Atia, and D. Iperciel. Educational data mining applications and tasks: A survey of the last 10 years. *Education and Information Technologies*, pages 1–17, jul 2017. ISSN 1360-2357.
- [3] G. Biswas, J. R. Segedy, and J. S. Kinnebrew. Smart open-ended learning environments that support learners cognitive and metacognitive processes. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7947 LNCS, pages 303–310, 2013. ISBN 9783642391453.
- [4] M. Cutumisu, K. P. Blair, D. B. Chin, and D. L. Schwartz. Assessing Whether Students Seek Constructive Criticism: The Design of an Automated Feedback System for a Graphic Design Task. *International Journal of Artificial Intelligence in Education*, 27(3):419–447, 2017. ISSN 1560-4306.
- [5] M. Desmarais and F. Lemieux. Clustering and visualizing study state sequences. *Proceedings of the 6th International Conference on Educational Data Mining (EDM 2013)*, pages 224–227, 2013.
- [6] K. E. DiCerbo and K. Kidwai. Detecting player goals from game log files. In *Proceedings of the 6th International Conference on Educational Data Mining, EDM 2013*, 2013. ISBN 9780983952527.
- [7] T. Dragon and B. P. Woolf. Guidance and Collaboration Strategies in Ill-defined Domains. *Workshop on ill-defined domains*, pages 65–73, 2006.
- [8] M. Easterday, D. Rees Lewis, and E. Gerber. Designing crowdcritique systems for formative feedback. *International Journal of Artificial Intelligence in Education*, 27(3):623–663, sep 2017. ISSN 1560-4292.
- [9] P. Fournier-Viger, R. Nkambou, and E. M. Nguifo. Building intelligent tutoring systems for ill-defined domains. *Studies in Computational Intelligence*, 308:81–101, 2010. ISSN 1860949X.
- [10] J. Gobert, M. Pedro, and R. Baker. From log files to assessment metrics for science inquiry using educational data mining. *Journal of the Learning Sciences*, 22:521–563, 01 2013.
- [11] N. Green. Argumentation scheme-based argument generation to support feedback in educational argument modeling systems. *International Journal of Artificial Intelligence in Education*, 27, 06 2016. doi: 10.1007/s40593-016-0115-y.
- [12] E. Harpstead and C. MacLellan. Investigating the Solution Space of an Open-Ended Educational Game Using Conceptual Feature Extraction. *Proceedings of the International Conference on Educational Data Mining*, 2013.
- [13] C. E. Hmelo-Silver, R. G. Duncan, and C. a. Chinn. Scaffolding and Achievement in Problem-Based and Inquiry Learning: A Response to Kirschner, Sweller, and Clark (2006). *Educational Psychologist*, 42(2):99–107, apr 2007. ISSN 0046-1520. doi: 10.1080/00461520701263368.
- [14] P. Jarušek, M. Klusacek, and R. Pelánek. Modeling Students' Learning and Variability of Performance in Problem Solving. *Proceedings of the 6th International Conference on Educational Data Mining*, pages 256–259, 2013.
- [15] D. H. Jonassen. Toward a Design Theory of Problem Solving, 2000. ISSN 09598049.
- [16] J. S. Kinnebrew, J. R. Segedy, and G. Biswas. Integrating Model-Driven and Data-Driven Techniques for Analyzing Learning Behaviors in Open-Ended Learning Environments. *IEEE Transactions on Learning Technologies*, 10(2):140–153, 2017. ISSN 19391382.
- [17] P. A. Kirschner, J. Sweller, and R. E. Clark. Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Educational Psychologist*, 41(2):75–86, 2006.
- [18] C. Lynch, K. Ashley, and T. Mitrovic. Intelligent Tutoring Technologies for Ill-Defined Problems and Ill-Defined Domains. *4th International Workshop on Intelligent Tutoring Systems and Ill-Defined Domains*, (Its), 2010.
- [19] C. F. Lynch, K. D. Ashley, V. Aleven, and N. Pinkwart. Defining "Ill-Defined Domains"; A literature survey. In *Intelligent Tutoring Systems*, 2006.
- [20] A. Mallavarapu, L. Lyons, E. Minor, B. Slattery, and M. Zellner. Developing Computational Methods to Measure and Track Learners' Spatial Reasoning in an Open-Ended Simulation. *Journal of Educational Data Mining (JEDM)*, 7(2):49–82, 2015. ISSN 2157-2100.
- [21] A. Mallavarapu, L. Lyons, S. Uzzo, W. Thompson, R. Levy-Cohen, and B. Slattery. Connect-to-Connected Worlds: Piloting a Mobile, Data-Driven Reflection Tool for an Open-Ended Simulation at a Museum. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, pages 1–14, 2019. ISBN 9781450359702. doi: 10.1145/3290605.3300237.
- [22] A. Mallavarapu, L. Lyons, Z. Elena, and S. Uzzo. Causal Modeling of Open-Ended Learning Environments for Generating Formative Feedback. In *Proceedings of the 2020 KDD Conference on Applied Data Science Track*, Under Review.
- [23] R. Martinez-Maldonado, K. Yacef, and J. Kay. Data Mining in the Classroom: Discovering Groups' Strategies at a Multi-tabletop Environment. *Proceedings of the 6th International Conference on Educational Data Mining*, pages 121–128, 2013.

- [24] J. Neubert, A. Kretschmar, and S. Greiff. Exploring exploration: Inquiries into exploration behavior in complex problem solving assessment. 07 2013.
- [25] A. Ogan, R. Wylie, and E. Walker. The challenges in adapting traditional techniques for modeling student behavior in ill-defined domains. *Intelligent Tutoring Systems for Ill-Defined Domains*, page 29, 2006.
- [26] I. Perikos, F. Grivokostopoulou, and I. Hatzilygeroudis. Assistance and feedback mechanism in an intelligent tutoring system for teaching conversion of natural language into logic. *International Journal of Artificial Intelligence in Education*, 27:475–514, 09 2017. doi: 10.1007/s40593-017-0139-y.
- [27] A. N. Rafferty, J. Davenport, and E. Brunskill. Estimating Student Knowledge from Paired Interaction Data. *Proceedings of the 6th International Conference on Educational Data Mining*, pages 260–263, 2013.
- [28] W. R. Reitman. Heuristic decision procedures, open constraints, and the structure of ill-defined problems. *Human judgments and optimality*, pages 282–315, 1964.
- [29] H. Schweigruber, T. Kelly, and H. Quinn. A framework for k-12 science education: Practices, crosscutting concepts, and core ideas. 2012.
- [30] V. J. Shute. Focus on Formative Feedback. *Review of Educational Research*, 78(1):153–189, 2008. ISSN 0034-6543.
- [31] H. A. Simon. The structure of ill-structured problems. 1973.
- [32] J. D. Slotta. Evolving the classrooms of the future: The interplay of pedagogy, technology and community. 2010.
- [33] R. J. Spiro, P. J. Feltovich, M. J. Jacobson, and R. L. Coulson. Cognitive flexibility, constructivism, and hypertext: Random access instruction for advanced knowledge acquisition in ill-structured domains. 2012.
- [34] B. Van de Sande. Applying Three Models of Learning to Individual Student Log Data. *Proceedings of the 6th International Conference on Educational Data Mining*, pages 193–199, 2013.
- [35] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

Extending the Hint Factory: Towards Modelling Productivity for Open-ended Problem-solving

Mehak Maniktala*
North Carolina State
University
mmanikt@ncsu.edu

Tiffany Barnes
North Carolina State
University
tmbarnes@ncsu.edu

Min Chi
North Carolina State
University
mchi@ncsu.edu

ABSTRACT

Determining when and whether to provide personalized support is a well-known challenge called the assistance dilemma. A core problem in solving the assistance dilemma is the need to discover when students are off-track or unproductive, so that the tutor can intervene. Such a task is particularly challenging for open-ended domains such as logic proofs, and programming. In this paper, we present a data-driven method to determine *step-level productivity* in a logic proof tutor. This approach leverages and modifies the Markov decision processes in the Hint Factory, a data-driven hint generator, to develop four productivity metrics. Our results provide evidence suggesting that, for each productivity metric, students' training productivity significantly correlates to their posttest performance. We conclude with a discussion outlining challenges posed when comparing these productivity metrics to a ground truth, and propose a preliminary approach to address them.

Keywords

productivity, open-ended, hint timing, assistance dilemma, logic proofs

1. INTRODUCTION

Intelligent Tutoring Systems (ITSs) provide individuals with adaptive feedback and hints, improving learning [25]. Studies suggest that hints, when provided appropriately, can augment students' learning experience [10, 22] and improve their performance [7]. However, researchers often find that students display poor help-seeking behavior [2, 21]; some abuse hints to expedite problem completion, and some avoid seeking help when they are in need [1, 20].

To deal with non-optimal help-seeking behavior, several ITSs provide unsolicited assistance [3, 18, 13]. However, determining *when* to provide proactive assistance, i.e., unsolicited

*This research was supported by the NSF grants 1726550 and 1651909.

Mehak Maniktala, Tiffany Barnes and Min Chi "Extending the Hint Factory: Towards Modelling Productivity for Open-ended Problem-solving" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 796 - 801

assistance in anticipation of future struggle, is particularly challenging in open-ended domains where there are many possible correct solutions. The assistance dilemma is a well-recognized challenge in the domain of ITSs, where a trade-off exists between giving and withholding information to achieve optimal learning [14]. On one hand, providing assistance may reduce frustration and save students' time, but may lead to shallow learning or a lack of motivation to learn by oneself. On the other hand, withholding information can encourage students to learn by themselves, but may lead to frustration and wasted time [14, 16]. A core problem of the assistance dilemma is the need to discover when students are off-track or unproductive so that the tutor can intervene. We hypothesize that developing a proactive hint policy for a logic tutor, where tutor interventions are delivered upon predictions of unproductivity in training, can improve students' logic proof strategies in a posttest without hints.

Contributions. In this paper, we present our novel, data-driven approach to measure productivity in an open-ended logic ITS. We extend the Hint Factory [24], a generalizable data-driven method for hint generation, to define four metrics of productivity. We then present our preliminary analysis on evaluating these metrics in the logic tutor.

2. RELATED WORK

Several studies have used the term "unproductive" to refer to undesirable behavior during training [12, 9, 19]. For example, Beck and Gong [8] define unproductive persistence or "wheel-spinning" based on whether or not a student achieved mastery (three correct problems) in 10 problem attempts. Their definition of unproductivity has been used in recent studies to predict when an intervention can help students by distinguishing between productive and unproductive behavior using decision trees [12] and Recurrent Neural Networks [9]. However, this definition of problem-level and problem-completeness based productivity is not suitable for our objective of guiding students toward efficient problem-solving strategies at a finer step-level granularity, specifically in open-ended domains.

McLaren et al. in a study on an open-ended inquiry-learning program defined unproductive events as the actions taken by the student that do not help them achieve the goal of a particular level, i.e., the steps that students take that are unlikely to advance their understanding of the concepts being taught [16]. Similar to this study, we define productivity on a step level to identify student steps that are not likely

to advance their problem-solving strategies. However, our definition is different in that we did not use a pre-defined domain-specific metric for efficient and inefficient strategies; rather we focused on solution length or optimality, which is valued across problem-solving domains.

In many multi-step open-ended but well-structured problem-solving domains, shorter solutions are considered to be more optimal than longer ones, and solving problems in less time reflects both learning and fluency [15, 23, 26, 11]. We use these basic assumptions about time and solution length to design a data-driven, domain-agnostic approach to model productivity on problem-solving steps. The Hint Factory is a data-driven approach for hint generation. The approach uses prior students' transaction log data to form an interaction network and assign scores to problem-solving states (snapshots of an on-going or completed proof) [24, 5]. A core insight of the work in this paper is that we can similarly use interaction networks to score productive problem-solving steps without the need to model the domain.

3. METHOD & PRELIMINARY RESULTS

In this section, we define *productivity*, a data-driven measure quantifying how much a student's most recent step contributes to an efficient solution. We explore four new productivity metrics defined using the combination of the *quality* of each state (local or global), and the amount of *progress* made in a step (absolute or relative). First, we present the Hint Factory. Next, we present our extension of the Hint Factory, and how we use it to define productivity.

3.1 Hint Factory

Hint Factory is a method for generating hints in multi-step open-ended domains [6]. In the Hint Factory, historical student solutions are used to form Markov decision processes (MDP) from interaction networks, where vertices are observed student problem-solving states (snapshots of their on-going or completed proof), and edges are problem-solving steps, i.e., a transition between states. The Hint Factory uses value iteration, a reinforcement learning technique, given in Equation 1, to assign an *expected value* $LQV(s)$ to each state s , where $LR(s)$ is the state's reward, γ is the discount factor, and $P_a(s, s')$ is the proportion of the observed solutions in state s that lead to state s' using the action (i.e. step) a . In the Hint Factory, a large reward is set for the problem-completion or *goal states* (100), penalties for incorrect states (10), and a cost for taking each action (1) [5]. A non-zero cost on actions causes the MDP to penalize longer solutions.

$$LQV(s) := LR(s) + \gamma \max_a \sum_{s'} P_a(s, s') LQV(s') \quad (1)$$

3.2 State Quality - Extending the Hint Factory metrics

In this section, we leverage the Hint Factory approach to generate two quality metrics that determine the expected values for each observed problem-solving state. The first metric of state quality was defined as part of our prior work on the Hint Factory, which we label as *local quality*. Local quality provides insights about how far a state is from the closest goal state, weighted by the probabilities of transitions, but it cannot provide information about whether the state is on an efficient path to a solution.

$$GQV(s) := GR(s) + \gamma \sum_{s'} P_a(s, s') GQV(s') \quad (2)$$

Global Quality. We devised a novel, data-driven global quality value function, GQV in Equation 2, to give higher values to states on efficient solution paths. Equation 2 sums $GQV(s')$ over all states s' reachable from s , weighted by $P_a(s, s')$, taking into account all future actions from a current state, rather than just the one with the max expected value. The global rewards GR are identical to LR for errors and actions, but are different for goals, giving shorter, more efficient solutions higher rewards. The global reward $GR(g)$ for each goal state g on a problem is $GR(g) = 100 - p * \delta(g)$ where $\delta(g)$ is the difference between the solution length of g and that of the shortest solution, and p is a penalty for longer solutions. We set $p = (100 - 80) / \delta_{median}$ where δ_{median} is the difference between the median and shortest solution lengths for each problem because median student solution lengths are assigned a global reward of 80. Meaning, the student's performance with a median solution length represents a low B grade. The proof of convergence for the modified value iteration equation 2 is given in appendix A.

We now demonstrate the differences between local and global quality metrics using solution trajectories (series of steps) of varying solution lengths: T_{short} , T_{medium} , and T_{long} in Figures 1 and 2. T_{short} is the shortest solution (four steps), with all nodes (logic statements) used. Note that a node is said to be *used* if it contributes towards deriving the conclusion of the problem. T_{medium} has five steps with one unused node D ; and T_{long} has eight steps, and all nodes used.

We generated interaction networks to determine the quality values for each problem using our historical data for $N = 796$ students. Figure 2 shows the quality values for the three trajectories in Figure 1. The start state in Figure 2 consists of the three given logic statements (the topmost state). Arrows between states represent steps, i.e., transition between states by logic rules applications. Non-start states are represented by a $+(XYZ)$, where XYZ is the new logic statement derived in a step. The start state has a high global quality, but low local quality. The start state's global quality is high because all efficient paths contain it, but its local quality is low because it is probabilistically farther away from goals than any other state in the figure. The local quality for states that are only found in incomplete attempts is lower than that for the start state. The local quality of the goal states on all three trajectories is 100. The global quality value for the goal state in each solution trajectory differs, with 100 for the T_{short} goal (since it's the most efficient), 95 for T_{medium} , and 80 for T_{long} goal states.

From the start state to the goal in T_{short} , both local and global quality state values increase monotonically since it is the most efficient solution. Note that not all quality values increase over every trajectory. For example, step $T_{medium} - 2$'s pre-state $(+A \rightarrow E)$ global quality is higher than that for its post-state $(+D)$ since the pre-state is on a more efficient path, but the local quality increases from pre- to post-state. Step $T_{long} - 3$'s pre-state $(+\neg E)$ has higher local and global quality values than its post-state $(+\neg E \rightarrow \neg A)$ since the post-state is farther from and less likely to reach T_{long} 's

Figure 1: Three Solutions with Varying Number of Steps for a logic problem in Deep Thought

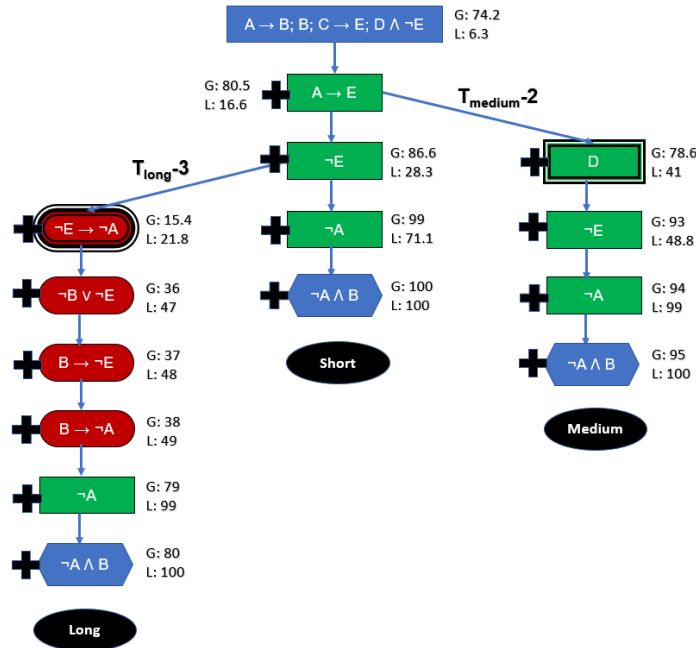
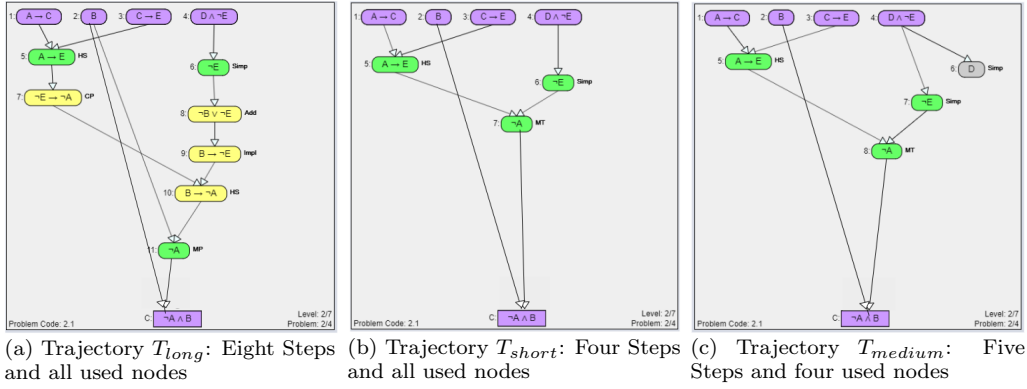


Figure 2: Illustration for the concepts of State Quality and Productive steps in 3 trajectories T_{short} , T_{medium} , and T_{long}

goal than the pre-state is to T_{short} 's closer goal. The global quality decreases for two reasons: (1) T_{long} goal is on a less efficient path, and (2) global quality performs a weighted sum over all the subsequent, previously-observed states in the larger (unshown) interaction network, many of which lead to incomplete attempts.

These examples demonstrate the differences between local and global state quality metrics. The main strength of generating these quality values is the MDP approach which ensures that each state quality value is based not only on the distance from a solution but also on the probability of transition at each of the successive steps. This allows us to rate steps in a more probabilistic manner than a simplistic comparison based on the distance from the most efficient expert solution.

3.3 Progress - Change in Quality

Since state quality is a measure of relative "goodness", we compare the quality of the current state with that of the previous and start states to evaluate the productivity in a step. In this section, we define two measures for *progress*: *relative*, the change in state quality from the *previous* problem-state, and *absolute*, the change in state quality from the *start*.

Relative progress is the difference between the quality values of the current and previous states. Relative progress with local quality identifies whether the previous or current state is closer to the goal. When using the global quality values, the relative progress identifies the state closer to a goal, and provides additional information detailing which one of the two is on a more efficient solution path.

Consider a valid, but long solution attempt. A relative progress measure reveals whether a student is progressing toward a solution in a step, but not whether their trajec-

Quality-Progress	Productivity Formula
Global-Absolute	$(GQV_{post-state} - GQV_{start-state}) \geq 0$
Global-Relative	$(GQV_{post-state} - GQV_{pre-state}) \geq 0$
Local-Absolute	$(LQV_{post-state} - LQV_{start-state}) \geq 0$
Local-Relative	$(LQV_{post-state} - LQV_{pre-state}) \geq 0$

Table 1: Step Productivity formula based on state Quality values and step Progress

tory is efficient. Therefore, we define *absolute progress* as the difference between the current and start states' quality, using either quality measure. Absolute progress using local quality reveals whether a student's current state is farther or closer from any goal states than when they began working on the proof. Global quality based absolute progress reveals the amount of efficient progress a student has made between the start and the goal. For example, if we compare the absolute progress on two consecutive steps of a solution attempt, if a student is taking efficient steps, then the absolute progress will increase on every step.

3.4 Productivity - Quality & Progress

We define four kinds of productivity measures based on quality {Local, Global} and progress {Relative, Absolute}. A step is considered *productive* if the progress of its post-state using either quality measure is a non-negative number, and *unproductive* otherwise, as shown in Table 1.

In consultation with an expert who has more than twenty years of experience teaching discrete math, we labeled the steps as productive or not in each of the three trajectories ($T_{short}, T_{medium}, T_{long}$) shown in Figure 1. Expert-assigned unproductive steps in Figure 2 are displayed in red and others are in green. According to the local quality and absolute progress (local-absolute) productivity metric, all the steps are productive because they eventually lead to a solution. However, this metric is not sensitive to variations in the solution lengths. When we use local-relative productivity, only the $T_{long} - 3$ step is unproductive, as it is the only step where a post-state is probabilistically farther from a solution than the pre-state. Using the global-relative measure, steps $T_{medium} - 2$ and $T_{long} - 3$ are unproductive because they have a pre-state on a more efficient path to the solution than the post-state. The global-absolute metric is the only measure that labels the four expert-identified unproductive steps correctly.¹ Note that each type of productivity captures a different perspective on a step towards the solution. Overall, the global-absolute productivity metric aligns perfectly with the expert's labels for the sample trajectories. However, using a panel of experts to rate each step would provide a more robust assessment of the ground truth. A major challenge in a manual inspection by a panel of experts is our vast state-space ($N = 72,560$).

3.5 Selecting a Productivity Metric

¹Note that these four unproductive states also correspond to the four infrequently used (yellow) nodes in the student solution shown in Figure 1a. However, some unproductive nodes have been observed to be frequently used, and some productive nodes to be infrequently used in our tutor, suggesting that the use-frequency alone cannot determine productivity

Productivity Using	Corr
Global-Absolute	0.328
Global-Relative	0.261
Local-Absolute	0.294
Local-Relative	0.236

Table 2: Correlation between students' Training Productivity and Posttest Optimality (all correlations are significant with $p < 0.01$)

To understand which one of the four productivity metrics is most indicative of how students' work in the tutor's training section affects their posttest solution optimality, we conducted a correlation test. Note, we evaluate students on an *optimality* score, which is as an exponential decay function on normalized steps e^{-steps} to account for the small variance in the number of steps. Steps are normalized to the interquartile range for each specific problem to account for varying lengths/difficulties. Very short solutions with step count less than or equal to Q1 (first quartile) have, *optimality* = 1, and those with step count greater than Q3 (third quartile) have an optimality score of 0.36 or less based on the exponential decay curve.

For each student in our dataset ($N = 437$), we computed their posttest optimality and the proportion of training steps that are productive using each productivity metric. We then calculated the correlation between each type of training productivity with posttest optimality using Pearson's coefficient. Table 2 shows that higher productivity in training is significantly correlated to better posttest optimality for all the productivity metrics. Among them, the global-absolute metric is the most correlated with posttest optimality.

4. CONCLUSION & ADVICE SOUGHT

In this paper, we provide a unique extension of the Hint Factory to determine productivity on a step-level in a logic ITS. Outside the scope of this paper, we assessed the impact of intervening with hints using a predictor of unproductivity (global-absolute) in a controlled study with two conditions: control and adaptive. Students in both conditions could request hints, but interventions using the predictor were given only in the adaptive condition. We found that the adaptive condition students had significantly better posttest optimality and time than their control peers. Our long term aim is to assess the generalizability of this approach in other open-ended domains such as programming, and to address the assistance dilemma for open-ended problem-solving.

For this doctoral consortium, we would like advice on how to further assess and compare the productivity metrics against a ground truth. We plan to form a panel of experts to rate a larger number of steps, but it is infeasible for them to rate each step in our vast state-space. Do we employ data-driven methods to determine the ground truth? I would like to discuss data-driven ways to evaluate the productivity metrics such as using inferring rewards from Gaussian processes [4] or using procedural solution generators [17].

Our preliminary results are promising, and through this consortium, we seek to determine a method to assess the ground truth of step-level productivity in the logic tutor.

5. REFERENCES

- [1] V. Aleven and K. R. Koedinger. Limitations of student control: Do students know when they need help? In *International Conference on Intelligent Tutoring Systems*, pages 292–303. Springer, 2000.
- [2] V. Aleven, B. McLaren, I. Roll, and K. Koedinger. Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education*, 16(2):101–128, 2006.
- [3] I. Arroyo, J. E. Beck, C. R. Beal, R. Wing, and B. P. Woolf. Analyzing students’ response to help provision in an elementary mathematics intelligent tutoring system. In *Papers of the AIED-2001 workshop on help provision and help seeking in interactive learning environments*, pages 34–46. Citeseer, 2001.
- [4] H. Azizsoltani, Y. J. Kim, M. S. Ausin, T. Barnes, and M. Chi. Unobserved is not equal to non-existent: using gaussian processes to infer immediate rewards across contexts. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 1974–1980. AAAI Press, 2019.
- [5] T. Barnes, J. Stamper, and M. Croy. „using markov decision processes for automatic hint generation “. *Handbook of Educational Data Mining*, 467, 2011.
- [6] T. Barnes, J. C. Stamper, L. Lehmann, and M. J. Croy. A pilot study on logic proof tutoring using hints generated from historical student data. In *EDM*, pages 197–201, 2008.
- [7] T. Bartholomé, E. Stahl, S. Pieschl, and R. Bromme. What matters in help-seeking? a study of help effectiveness and learner-related factors. *Computers in Human Behavior*, 22(1):113–129, 2006.
- [8] J. E. Beck and Y. Gong. Wheel-spinning: Students who fail to master a skill. In *International conference on artificial intelligence in education*, pages 431–440. Springer, 2013.
- [9] A. Botelho, A. Varatharaj, T. Patikorn, D. Doherty, S. Adjei, and J. Beck. Developing early detectors of student attrition and wheel spinning using deep learning. *IEEE Transactions on Learning Technologies*, 2019.
- [10] A. Bunt, C. Conati, and K. Muldner. Scaffolding self-explanation to improve learning in exploratory learning environments. In *International Conference on Intelligent Tutoring Systems*, pages 656–667. Springer, 2004.
- [11] H. Cen, K. R. Koedinger, and B. Junker. Is over practice necessary?-improving learning efficiency with the cognitive tutor through educational data mining. *Frontiers in artificial intelligence and applications*, 158:511, 2007.
- [12] S. Kai, M. V. Almeda, R. S. Baker, C. Heffernan, and N. Heffernan. Decision tree modeling of wheel-spinning and productive persistence in skill builders. *JEDM Journal of Educational Data Mining*, 10(1):36–71, 2018.
- [13] S. Kardan and C. Conati. Providing adaptive support in an interactive simulation for learning: An experimental evaluation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3671–3680. ACM, 2015.
- [14] K. R. Koedinger and V. Aleven. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, 19(3):239–264, 2007.
- [15] R. E. Mayer. *Thinking, problem solving, cognition*. WH Freeman/Times Books/Henry Holt & Co, 1992.
- [16] B. M. McLaren, M. Timms, D. Weihnacht, D. Brenner, K. Luttgen, A. Grillo-Hill, and D. H. Brown. A web-based system to support inquiry learning. In *Proceedings of the 6th International Conference on Computer Supported Education-Volume 1*, pages 43–52. SCITEPRESS-Science and Technology Publications, Lda, 2014.
- [17] K. L. McMillan and A. Rybalchenko. Solving constrained horn clauses using interpolation. *Tech. Rep. MSR-TR-2013-6*, 2013.
- [18] R. C. Murray and K. VanLehn. A comparison of decision-theoretic, fixed-policy and random tutorial action selection. In *International Conference on Intelligent Tutoring Systems*, pages 114–123. Springer, 2006.
- [19] S. Park and N. Matsuda. Predicting students’ unproductive failure on intelligent tutors in adaptive online courseware. In *Proceedings of the Sixth Annual GIFT Users Symposium*, volume 6, page 131. US Army Research Laboratory, 2018.
- [20] T. W. Price, Z. Liu, V. Cateté, and T. Barnes. Factors influencing students’ help-seeking behavior while programming with human and computer tutors. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*, pages 127–135. ACM, 2017.
- [21] T. W. Price, R. Zhi, and T. Barnes. Hint generation under uncertainty: The effect of hint quality on help-seeking behavior. In *International Conference on Artificial Intelligence in Education*, pages 311–322. Springer, 2017.
- [22] M. Puustinen. Help-seeking behavior in a problem-solving situation: Development of self-regulation. *European Journal of Psychology of education*, 13(2):271, 1998.
- [23] M. U. Smith. *Toward a unified theory of problem solving: Views from the content domains*. Routledge, 2012.
- [24] J. Stamper, T. Barnes, L. Lehmann, and M. Croy. The hint factory: Automatic generation of contextualized help for existing computer aided instruction. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems Young Researchers Track*, pages 71–78, 2008.
- [25] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.
- [26] K. Yacef. The logic-ita in the classroom: a medium scale experiment. *International Journal of Artificial Intelligence in Education*, 15(1):41–62, 2005.

APPENDIX

A. PROOF OF CONVERGENCE FOR THE MODIFIED BELLMAN BACKUP FUNCTION

Theorem: The modified Value iteration (Eqn 2) converges to GQV^* for any initial estimate GQV , i.e.,

$$\lim_{k \rightarrow \infty} GQV_k = GQV^* \quad \forall GQV$$

For any estimate of the value function GQV , we define the modified Bellman backup operator $\hat{B} : R^{|S|} \rightarrow R^{|S|}$

$$\hat{B}GQV(s) = GR(s) + \gamma \sum_{s' \in S} P_a(s, s') GQV(s')$$

Before we provide the proof of convergence, we provide the proof of contraction, i.e, for any two value functions GQV and GQV' :

$$\|\hat{B}GQV_k - \hat{B}GQV'_k\| \leq \gamma \|GQV_k - GQV'_k\|$$

where the max norm:

$$\|GQV\| = \max_{s \in S} |GQV(s)|$$

$\|v - v'\|$ = Infinity norm (max difference over all states)

Proof of contraction:

$$\|\hat{B}GQV - \hat{B}GQV'\|$$

$$= \left\| \left[GR(s) + \gamma \sum_{s' \in S} P_a(s, s') GQV(s') \right] - \left[GR(s) + \gamma \sum_{s' \in S} P_{a'}(s, s') GQV'(s') \right] \right\|$$

$$= \gamma \left\| \left[\sum_{s' \in S} P_a(s, s') GQV(s') - \sum_{s' \in S} P_{a'}(s, s') GQV'(s') \right] \right\|$$

$$= \gamma \left\| \left[\sum_{s' \in S} P_a(s, s') (GQV(s') - GQV'(s')) \right] \right\|$$

$$\leq \gamma \max_s \sum_{s' \in S} P_a(s, s') |GQV(s') - GQV'(s')|$$

$$\leq \gamma \sum_{s' \in S} P_a(s, s') \|GQV - GQV'\|$$

$$= \gamma \|GQV - GQV'\|$$

since $P_a(s, s')$ are non-negative and sum to one

Proof of Convergence:

$$\begin{aligned} & \|GQV_{k+1} - GQV^*\|_\infty \\ &= \left\| \hat{B}GQV_k - GQV^* \right\|_\infty \\ &\leq \gamma \|GQV_k - GQV^*\|_\infty \leq \dots \\ &\leq \gamma^{k+1} \|GQV_0 - GQV^*\|_\infty \rightarrow 0 \end{aligned}$$

Scalability in Online Computer Programming Education: Automated Techniques for Feedback, Evaluation and Equity

Jessica McBroom, Kalina Yacef and Irena Koprinska
School of Computer Science, University of Sydney, Sydney, NSW 2006, Australia
{jmc6755, kalina.yacef, irena.koprinska}@sydney.edu.au

ABSTRACT

The delivery of programming courses online offers great promise to provide quality programming education in an accessible manner. However, it also introduces new challenges, including how to maintain course quality as the ratio of students to teaching staff increases. In particular, the provision of effective feedback, detailed course evaluations and the promotion of equity can all become more challenging as the size of a course increases. This work explores, integrates and develops potential data mining and artificial intelligence techniques that could be utilised to address these issues in the context of programming education.

Keywords

programming education, automated feedback, course evaluation, equity, computer science education, student behaviour

1. INTRODUCTION

In recent decades, online programming courses have become increasingly numerous, with a diverse range of languages being taught on a variety of platforms. Due to their online nature, these courses are highly accessible and available to a large number of students. In addition, they do not require teachers and students to be in close proximity, making them more robust to disruptions from global events, such as the COVID19 pandemic [1]. Considering these benefits, methods for increasing the effectiveness of these courses are of great significance, with the potential to benefit many thousands of students.

One important challenge in educational settings is ensuring there is effective information flow between teachers and students. In particular, there should firstly be a way for instruction and feedback to flow from teachers to students so that students can learn. In addition, there should also be a way for information about student understanding and progress to flow from students to teachers so that teachers can adapt and improve the instruction and feedback (re-

ferred to as “closing the loop” [3]). This challenge becomes increasingly significant in the context of online education, where teachers and students may not be in direct contact and there may be a large number of students per teacher.

Some approaches to improving information flow in online education have focused on opening communication channels between teachers and students. For example, discussion boards [16], one-on-one chats with tutors [7] and student surveys [17] all allow teachers to provide instruction to or receive feedback from students. One issue with these approaches, however, is that they can suffer from scalability issues. For example, as the ratio of students to teachers increases, it becomes more difficult for teachers to monitor every discussion board question, to engage in every chat or to carefully read all open-ended survey responses. In addition, approaches to dealing with this, such as making surveys quantitative, can limit the detail of the feedback.

Since direct communication channels for closing the loop can suffer from scalability issues, another approach is to introduce automated systems as interfaces between teachers and students. In particular, teachers can configure an automated system for providing instruction and feedback to students, thereby allowing information to flow from teachers to students. In addition, automated systems can be used to analyse student behaviour and report back information about student progress to teachers, which they can use to re-configure the system, forming a loop as shown in Figure 1. This model can then be extended to consider additional loops, such as between students and the feedback system. Such systems are a highly promising approach to providing scalable education to students.

This work focuses on addressing a few key challenges associated with this approach that are both particularly important in the context of programming education and also less developed in the field at large. In particular, it first considers automated feedback techniques, and how these can be integrated together to gain a coherent picture of the current state of the field. In addition, it develops new EDM techniques for analysing student behaviour in order to evaluate a course generally, and also in the context of equity. As such, this work acts as a step towards improving the scalability and effectiveness of online programming education.

2. CONTRIBUTIONS

2.1 HINTS: A Framework for Automated Hints

Jessica McBroom, Kalina Yacef and Irena Koprinska "Scalability in Online Computer Programming Education: Automated Techniques for Feedback, Evaluation and Equity" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 802 - 805

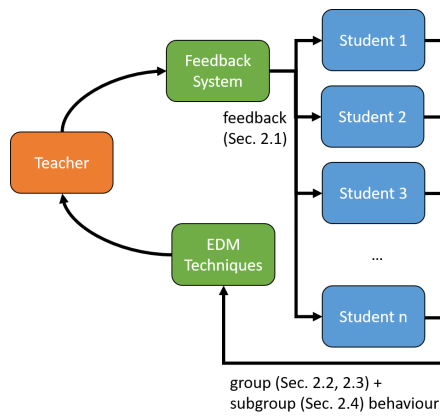


Figure 1: An example of scalable information flow between teachers and students. First, the teacher configures a system to provide automated feedback to students. Then, EDM techniques are used to analyse the behaviour of students and inform the teacher of their progress, thereby “closing the loop”. The teacher can then make improvements to the system based on this information. The section labels indicate how the work described in the next section relates to this model.

When undertaking a programming course, students are often presented with various programming exercises to complete in order to gain practical experience. Since students commonly find these tasks challenging, an important component of an automated feedback system is the ability to provide hints to students experiencing difficulty on programming tasks. Such hints could include suggestions about where the student went wrong, how to proceed or concepts to revise.

A wide range of interesting techniques have been developed to provide automated hints to students, including hints that use data from peers, model solutions and test cases and utilise a diverse range of methods to produce hints, including neural networks [2], Markov Decision Processes [15], program synthesis techniques [6] and a variety of other approaches [4, 5, 13]. In addition, systems employing automated hint techniques differ with respect to the programming language taught, evaluation method, student cohort and learning context. While this diversity offers great promise for producing high quality hints and represents the great interest in the area, it also increases the difficulty of understanding the types of techniques that are available and how they fit together.

The first contribution of this work, described in detail in [9], is a framework and survey to draw these techniques together into a coherent picture. In particular, the contribution is the *Hint Iteration by Narrow-down and Transformation Steps* (HINTS) framework. This framework focuses on understanding hint techniques by decomposing the process they use to produce hints into a series of smaller steps, which can then be fit into two categories: narrow-down and transformation steps. For example, in [14] the first step is to find the closest solution to a student’s program, and the second step is to then find edits from the student’s program to that solution. Once hint techniques are considered as a series of smaller steps, it becomes much easier to relate techniques

based on their steps to see how they fit together and the potential avenues for further developing them.

A summary of the five most important insights gained by surveying automated hint techniques in this manner are as follows:

1. Even if hint techniques appear very different overall, they can be related together by their smaller steps. This allows, for example, for them to be collected together into a single diagram to provide a coherent picture of the current state of hint techniques.
2. It is important to consider individual steps of hint techniques when evaluating, discussing and developing hint techniques, since these can often be mixed and matched.
3. there appears to be a theoretically motivated relationship between some hint technique steps and data-driven evaluation techniques, which would be interesting to explore further
4. The question of why hint techniques can be fit together in this way provides insight into the nature of automated hint generation. In particular, the fact that hint techniques can be decomposed into a series of smaller steps that fit into only two simple categories suggests this structure may be necessitated by the problem in general.
5. Further work on hint technique evaluation methods is necessary - there are so many possible combinations of steps that the efficiency of evaluation techniques must be improved.

Since understanding the types of hint techniques that are available and how they fit together is an important step in developing and evaluating automated feedback systems, this work contributes to improving such systems, which are an important component of scalable programming education.

2.2 A Technique for Clustering Student Programs

An important aspect of understanding student learning in a programming course is the ability to visualise the behaviour of students on programming exercises. In particular, understanding how beginner students behave during the first few exercises is of particular importance, since these students are more likely to make many mistakes, be least equipped to correct these mistakes and potentially be discouraged from the area if they experience too much difficulty.

The second contribution of this work is a technique for clustering beginner student programs in order to visualise trends in student behaviour when completing programming tasks. This technique, described in more detail in [12], involves first applying transformations to group logically equivalent programs. The resulting groups are then further combined if the structures of the programs in the groups are the same up to a customisable threshold and they pass the same test cases. In this way, programs with a similar functionality are clustered together, and it is possible to understand all possible programs in a cluster using a single sample program from the cluster.

After the clustering is performed, the clusters can be organised into a network with edges showing how students transition between clusters as they work on an exercise. Sequential pattern mining can also be performed to discover the most frequent transitions. This can then reveal information about where students experience difficulty (indicated by loops in the network) and the strategies they follow to complete an exercise. Example applications are also discussed in [12].

Planned extensions of this work include a more thorough evaluation beyond the case study in [12], and also an exploration of this technique in the context of equity, as discussed in Section 2.4.

2.3 DETECT: A General Clustering Technique for Temporal Trends in Student Behaviour

Many EDM techniques utilise clustering to understand student behaviour since clustering can condense highly complex information into simpler and more manageable subsets. This can allow the results to be more easily interpreted and thereby provide greater insight into student behaviour. However, while existing techniques can be readily applied to discover different types of student behaviour, it can often be more challenging to use them to discover particular behavioural trends in time. This is because the objective functions they use, which guide the cluster formation, often do not consider temporal information.

The third contribution of this work is DETECT (*Detection of Educational Trends Elicited by Clustering Time-series data*), a customisable hierarchical clustering algorithm for detecting trends in student behaviour over time. This algorithm, described in detail in [11], produces clusters similar in structure to a decision tree, with clusters defined by decision rules (e.g. a cluster may be all examples where the time taken was ≤ 5 mins and the student's grade was A). To form these clusters, DETECT uses a customisable objective function, which governs the types of temporal trends that are found. For example, these could include behavioural changes between the start and end of a course, or behaviours that make an exercise stand out from the ones before and after it. The algorithm then works by recursively dividing the examples into subsets in the manner that maximises the objective function.

Some advantages of DETECT in an educational context are as follows:

1. It is applicable to a wide range of educational datasets. A core feature of educational courses is that they tend to have a repeating structure. For example, they may have a series of lectures, homework tasks, assignments, tutorials, readings or practice exercises, which each have a similar structure. As such, these courses can be divided into time steps with similar features, which is the type of data DETECT is applicable to. For example, time steps could be homework tasks and features could include the time taken and grade. This allows DETECT to be applied to a wide range of educational data set, including programming data.
2. The objective function is customisable and has few constraints. In [11], two different objective function examples are given: one for finding differences between

the start and end of a course, and one for finding behaviours that characterise a particular exercise. However, this function can be customised to find other types of trends with minimal constraints (e.g. the function doesn't need to be differentiable). This allows DETECT to be highly flexible.

3. The results are easy to interpret. Since the clusters are defined by decision rules, it is easy to understand exactly which examples belong to each cluster, thereby improving the interpretability of the results
4. The algorithm is more robust to dependencies between features than traditional clustering methods, since the objective function can use temporal information to evaluate cluster quality. In particular, DETECT places higher weight on features that reveal interesting trends in time beyond what has already been found. As such, highly correlated features are penalised, making DETECT more robust to dependencies between features.

Planned extensions of this work include a deeper exploration into potential objective functions beyond the two discussed in [11] and a more thorough evaluation of the algorithm.

2.4 Adapting Techniques to Explore Equity Issues

One important issue when applying data mining techniques to student data is that patterns from under-represented groups can sometimes be obscured by collective trends. For example, if clustering is used to find the general types of student behaviour overall, this may obscure the potentially unique behaviour of subgroups, especially if they are small. As such, an important aspect of closing the loop between teachers and students is ensuring that analysis techniques not only provide information about the majority of students, but also minority groups.

The fourth contribution of this work is an exploration of equity issues in the context of programming education, and how the proposed techniques might be adapted to provide information about under-represented student groups. This work is still in progress, but so far has included:

1. an exploration of gender differences in enrolment and exercise completion rates in a series of programming courses, described in [10]. In particular, the courses were run for school students during a 5 week Python programming challenge in Australia in 2018, and included both block-based and text-based courses of different difficulty levels. In general, there were approximately twice as many male enrolments as female, but little difference in exercise completion rates between genders. Such an analysis acts not only as an important first step in understanding the nature of student differences in a course, but also as a baseline with which to compare data mining techniques
2. adapting DETECT to consider gender and school grade differences among students in the lead up to them dropping out. [8] In particular, this involved selecting 10 evenly spaced out programming exercises for each student who dropped out (i.e. the first exercise they completed, the last exercise they completed and

8 equally spaced out exercises in between). These exercises could then be used as time periods that were relative to the students (i.e. time 1 was when he student first began, time 2 was 10% of the way through their interaction with the course, and time 10 was the last exercise they completed before dropping out). DETECT could then be applied to this data to see the largest changes in student behaviour approaching dropping out, which could then be filtered based on gender and school grade to observe differences in the lead up to dropout for different groups.

Planned extensions to this work include adapting the program clustering technique from Section 2.2 to consider equity, and to extend the analyses by considering data from consecutive years.

3. CONCLUSION

In the context of programming education, one approach to increasing information flow between teachers and students in a scalable manner is to introduce automated systems as interfaces between teachers and students. This work aims to address a few key theoretical challenges in working towards this, with a focus on the automated techniques necessitated by such an approach. In particular, it presents a framework for integrating automated programming hint techniques, two clustering techniques for analysing student behaviour and an exploration of how such techniques can be adapted to investigate equity issues. As such, it acts as a step towards increasing information flow between teachers and students in a scalable manner, with the ultimate aim of improving programming education.

4. ADVICE SOUGHT

Any general feedback on this work, including its contributions and the problems it addresses would be most welcome, particularly in the context of thesis writing. Additionally, any suggestions for improving the coherence or completeness of the work, or other ideas for improvement would be of great value. For context, this work has been conducted over 2.5 years of PhD study, with up to 1.5 years remaining.

5. REFERENCES

- [1] 290 million students out of school due to covid-19: Unesco releases first global numbers and mobilizes response. <https://en.unesco.org/news/290-million-students-out-school-due-covid-19-unesco-releases-first-global-numbers-and-mobilizes>. Accessed: 14/04/2020.
- [2] S. Bhatia and R. Singh. Automated correction for syntax errors in programming assignments using recurrent neural networks. *arXiv preprint arXiv:1603.06129*, 2016.
- [3] D. Clow. The learning analytics cycle: closing the loop effectively. In *Proceedings of the 2nd international conference on learning analytics and knowledge*, pages 134–138, 2012.
- [4] B. Edmison and S. H. Edwards. Applying spectrum-based fault localization to generate debugging suggestions for student programmers. In *2015 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pages 93–99. IEEE, 2015.
- [5] A. Gerdes, B. Heeren, J. Jeuring, and L. T. van Binsbergen. Ask-elle: an adaptable programming tutor for haskell giving automated feedback. *International Journal of Artificial Intelligence in Education*, 27(1):65–100, 2017.
- [6] A. Head, E. Glassman, G. Soares, R. Suzuki, L. Figueredo, L. D’Antoni, and B. Hartmann. Writing reusable code feedback at scale with mixed-initiative program synthesis. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, pages 89–98, 2017.
- [7] B. Jeffries, T. Baldwin, M. Zalk, and B. Taylor. Online tutoring to support programming exercises. In *Proceedings of the Twenty-Second Australasian Computing Education Conference*, pages 56–65, 2020.
- [8] J. McBroom, I. Koprinska, and K. Yacef. How does student behaviour change approaching dropout? a study of gender and school year differences. *Unpublished, accepted at EDM2020*.
- [9] J. McBroom, I. Koprinska, and K. Yacef. A survey of automated programming hint generation - the HINTS framework. *arXiv preprint arXiv:1908.11566 (Unpublished, submitted to ACM Computing Surveys)*, 2019.
- [10] J. McBroom, I. Koprinska, and K. Yacef. Understanding gender differences to improve equity in computer programming education. In *Proceedings of the Twenty-Second Australasian Computing Education Conference*, pages 185–194. ACM, 2020.
- [11] J. McBroom, K. Yacef, and I. Koprinska. DETECT: A hierarchical clustering algorithm for behavioural trends in temporal educational data. *arXiv preprint arXiv:2005.10640 (Unpublished, accepted at AIED2020)*.
- [12] J. McBroom, K. Yacef, I. Koprinska, and J. R. Curran. A data-driven method for helping teachers improve feedback in computer programming automated tutors. In *International Conference on Artificial Intelligence in Education*, pages 324–337. Springer, 2018.
- [13] B. Paaßen, B. Hammer, T. W. Price, T. Barnes, S. Gross, and N. Pinkwart. The continuous hint factory-providing hints in vast and sparsely populated edit distance spaces. *arXiv preprint arXiv:1708.06564*, 2017.
- [14] T. Price, R. Zhi, and T. Barnes. Evaluation of a data-driven feedback algorithm for open-ended programming. In *Proceedings of the 10th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2017.
- [15] T. W. Price, Y. Dong, and T. Barnes. Generating data-driven hints for open-ended programming. In *Proceedings of the 9th International Conference on Educational Data Mining*. International Educational Data Mining Society, 2016.
- [16] J. Suler. In class and online: Using discussion boards in teaching. *CyberPsychology & Behavior*, 7(4):395–401, 2004.
- [17] S. Watson. Closing the feedback loop: Ensuring effective action from student feedback. *Tertiary education and management*, 9(2):145–157, 2003.

Investigating Students' Learning in Online Learning Environment

Lavendini Sivaneasharajah, Katrina Falkner, Thushari Atapattu
The University of Adelaide

{lavendini.sivaneasharajah, katrina.falkner, thushari.atapattu}@adelaide.edu.au

ABSTRACT

Due to the increasing interest in the online learning environment, particularly in Massive Open Online Courses (MOOCs), predictions and education data mining have rapidly gained prominence in education studies over the past decade. The massive amount of student data available in MOOC platforms enables us to gain insight into students' learning behaviours. Therefore, this paper outlines the doctoral work that explores the idea of 'student roles' and their linguistic changes to analyse the students' learning behaviours in MOOCs. A multi-class classifier has been built to identify user roles (e.g. information seeker, information giver) with 82.30% F-measure. Preliminary results on linguistic experiments demonstrate, distinguish linguistic behaviours can be observed in different user roles. The outcome of this research study will contribute to a learning model that can be used to understand students' learning process.

Keywords

MOOCs, Discussion forums, User Role, Natural Language Processing, Machine Learning.

1. INTRODUCTION

Learning Analytics has been gaining high popularity in recent years among research scholars due to the challenges it imposes; two of which are increasingly complex, large-volume data and heterogeneous data. Integrating several sources of data that are generated during learning activities is of major need for the education sector to provide timely enhanced services to both students and instructors. Integrating and analysing student data can contribute immensely towards reducing dropout rates, timely instructor interventions, and many other [4].

In the 21st century, students are more exposed to Massive Open Online Courses (MOOCs) and online learning environments as they believe it is more beneficial than the traditional learning environment such as flexible study hours, availability for everyone [7]. As many of the MOOCs are freely available for students, it draws the interest of thousands of learners. However, accessing the success rate of a student learning in online platforms has become difficult as students enrol for varying purposes. Knowing that students may enrol in courses for other purposes, we need to explore other perspectives of learning success beyond

completion.

MOOC contains many types of resources to support students in their learning activities. These elements can be categorised as videos, lecture series, reading materials, quizzes, assignments, discussion forums etc. According to Anderson [1], discourse enables the learner to come up with their own reasoning and logical thinking by communicating with others. Thus, investigating discussion forums will help researchers to understand the actual situation of the students in the learning lifecycle.

The overarching aim of this doctoral work is to understand students' learning with time within MOOCs. To this end, the research mainly focuses on examining user role transformation and linguistic change that occurs in discussion forums with time. We believe analysing these roles and associated linguistic changes will eventually result in a deeper understanding of the student's learning lifecycle. Further, this research will also investigate the influencing factors (e.g. course structure, learners' demographic) that influence these observable features (i.e. student role, linguistic expression) and their correlations.

To achieve the aforementioned aim, our investigations are driven by two main research questions (RQ):

RQ 1: Can student role and linguistic expressions be used to understand student learning? (1. How to build a predictive model that predicts students' roles in an online learning environment? 2. How to track the linguistic change of each student roles in the online learning environment?).

RQ 2: To what extent students' learning is affected by external factors? (1. What are the external factors that affect these transformations (user role and linguistic change)? 2. What are the correlations between external factors and these transformations?).

The contribution of this doctoral work includes a predictive model that leverages linguistic-only features to predict student roles in discussion forums. Further, this doctoral work develops a linguistic framework to understand students' learning. This demonstrates distinguish linguistic behaviours of different student clusters in discussion forums. Moreover, identifying the external factors such as course structure, learners' demographic that affect students' learning and their correlation.

2. RELATED WORK

2.1 Post classification and role identification in discussion forums

User role classification is grounded by post-classification methodologies that prevail in the existing literature. In other words, post-classification is the foremost step that needs to be carried out in order to identify the user roles in discussion forums. With the examinations on speech acts by Searle [9], there are

Lavendini Sivaneasharajah, Katrina Falkner and Thushari Atapattu "Investigating Students' Learning in Online Learning Environment" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 806 - 809

several post-classification methodologies have been introduced to the research community.

While prior studies classify forum posts into different categories such as question, answer, solutions, Hecking et al. [6] have carried out post-classification by generalising the categories that prevail in the existing studies [2]. The study presents three different classes namely: information seeking, information giving and other. Hecking et al. [6] achieved 70% accuracy using content-related features (e.g. phrases – “need help or helps you”) and contextual features (e.g. position in the thread, number of votes) for classification purposes. However, relying on contextual features for forecasting is not feasible in a real-time system as these contextual features changes with time. And predictions can only be made at the end of the course as they occur during the course. Therefore, our study aims to build a predictive model for discussion forum classification using linguistic-only features while eliminating contextual and structural features.

2.2 Linguistic change in online communities

For decades, researchers in the linguistic discipline have explored the language change in many different spheres starting from historical linguistics to sociolinguistic. Research scholars believe exploring temporal changes in user’s language will provide useful insights to research communities. Linguistic research has taken various paths with time to exhibit correlations between the linguistic and other aspects such as historical change, community norms, user lifespan etc.

The work by Nguyen et al. [8] identifies the relationship between community membership and language use. According to their findings, forum specific jargons and informal linguistic style can be observed in long-term participants’ discourse. Dowell et al. [5] have conducted a study on MOOC data to identify the conversion in learner’s language and discourse characteristic with time. However, the research did not investigate the linguistic changes associated with each user role. It is said that learner’s language changes with time, especially discourse in discussions forums will be topic-oriented and reflective of deep learning with the consequent offerings of a course [5]. Nevertheless, investigating linguistic change for a student role has not been addressed. Even though preliminary work on linguistic change has been conducted in other online communities, there is a lack of work conducted in MOOCs.

3. METHODOLOGY

This doctoral work will be conducted in two main phases namely:

1. Pilot study - Identifying potential features from discussion forums.
2. Building machine learning model - Implementing a model to understand student learning. The phase two will be further divided into three sub tasks to address aforementioned research questions as follows:

Task 1: Developing predictive model to identify user roles (IG/IS and O) in discussion forums.

Task 2: Developing a machine learning model to track linguistic change.

Task 3: Identifying external factors and their correlations with user roles and linguistic expressions.

4. EXPERIMENTS AND RESULTS

The study collected 9,497 user posts from 923 users from the AdelaideX¹ ‘Introduction to Project Management’ and ‘Risk Management for Projects’ courses offered in 2016 and 2017 respectively. The current study was conducted using 6000 posts from ‘Introduction to Project Management’. Two independent human evaluators carried out a manual annotation with a high inter-rater agreement (Cohen’s kappa = 0.925) and annotated the user posts as information seeker (IS), information giver (IG) and other (O).

4.1 How to build a predictive model that predicts students’ roles in an online learning environment? (RQ1)

A multi-class classifier was built to predict user roles (IG, IS and O) for a given forum post using discourse features and linguistic features. The features were extracted using Pennebaker’s Linguistic Inquiry and Word Count (LIWC) tool² which generates different linguistic measures for an input text. The study selected sixteen optimal features using Recursive Feature Elimination with Cross-Validation feature selection technique.

We implemented following multiclass classifiers with different sets of algorithms using Weka: Naïve Bayes, Random Forest, Simple Logistic Regression, Logistic Regression and Sequential Minimal Optimisation (SMO). All these classifiers were tested using 10 Fold Cross-Validation to assess the accuracy. Among these, the Random Forest classification model performed best with 82.30 of F measure.

Further, we also fine-tuned the parameters for Random Forest classifier using the scikit-learn library (RandomizedSearchCV and GridSearchCV). The results show that Random Forest classifier performs at its best in the following parameter setting: n_estimators:400, ‘min_samples_split’:10, ‘min_samples_leaf’: 4 and max_depth: 70.

Table 1: Results of classifier performance

Classifiers	Accuracy	Precision	Recall	F1	Cohen’s Kappa
Naïve Bayes	71.28	74.40	71.30	71.00	0.5117
Random Forest	82.17	82.30	82.20	82.20	0.6955
Simple Logistic	79.35	79.60	79.40	79.40	0.6473
Logistic	79.43	79.70	79.40	79.50	0.6498
SMO	74.80	76.50	74.80	75.30	0.5770

The work by Hecking et al. [6] is the only existing work in our workspace that classify the discussion forums posts as information giving, information seeking and other. They have achieved an overall of 71.5 F-measure for IS and IG class while obtained an average of 70 and 66 for precision and recall respectively across all three classes. With 82.30% of F-measure, we have demonstrated that analysing the language of post content itself is sufficient to predict user roles. Therefore, it is evident that linguistic features have a high impact on user role prediction in discussion forums.

¹ <https://www.edx.org/school/adelaideX>

² <https://liwc.wpengine.com/>

4.2 How to track the linguistic change of each student roles in the online learning environment?

An important component of this research study is to propose a linguistic framework that can exhibit the linguistic characteristic of different student clusters that will be identified by this study. Afterwards tracking their changes associated with each student role. To achieve this, we carried out linguistic experiments to identify distinguishing characteristics between these student roles.

We started with a simple word count and performed the one-way analysis of variance (ANOVA) for different student roles. The analysis shows that the mean value of information giver is higher than information seeker, and there is a significant difference between the mean values ($p < 0.005$). The results indicate that information giver tend to use more words when reflecting their thoughts than the information seeker in discussion forums.

Using a similar approach, we computed the frequency of n-grams in given user posts. We created a vocabulary list using n-grams from lecture transcripts. Then, the study computed the lexical frequency profile (LFP) for each user role. We created a Phrase Matcher Object and applied the matcher object on each user post to extract the keywords. For a given user, the number of keywords used in information giving post increases – reach an optimal number and decreases with time, whereas for information seeking post it increases/decreases with time – also, a minimum level of changes observed in other user post. Moreover, information giver uses more keywords from the lecture transcript than information seeker and other. Further, there is a considerable amount of drop-in ‘other’ user role. Figure 1 shows the Lexical Frequency Profile for a sample of five users with time.

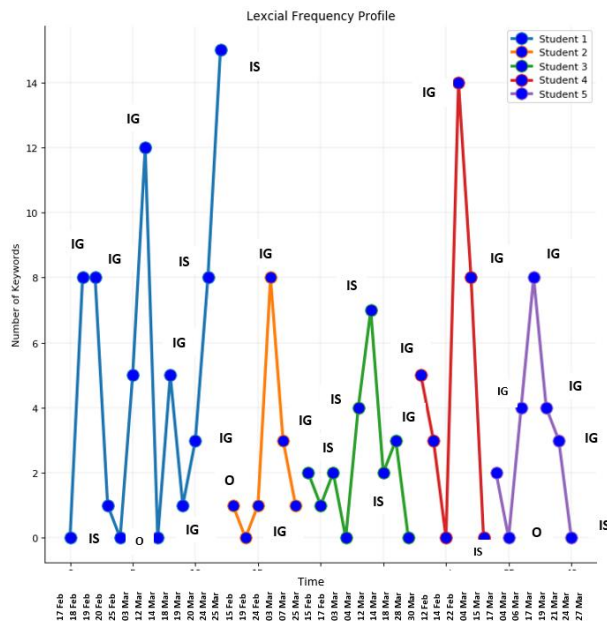


Figure 1: Lexical frequency profile across user role

Information embeddedness is one of the key elements that contribute toward student learning. This study attempts to find the level of information embeddedness using clause extraction. Clause extraction has been used to determine the relationship between the clauses per sentence and language development. We

develop a novel approach in which clauses have been extracted from the parse tree using a rule-based approach. A pipeline is being built with Part-Of-Speech (POS) tagging using Stanford CoreNLP³ to get the basic interpretation of a student post. Tree Annotation is used to extract a parse tree for a given sentence. Initially, clause-level tags (e.g. SBAR) and word-level coordinating conjunction (e.g. CC) have been extracted from the parse tree. Then, we implemented a rule-based approach to extract the number of clauses.

According to Crossley et al. [3], discourse complexity can be measured by any given reading level measures. Therefore, we used Flesch-Kincaid reading level measure to explore discourse complexity with time for each user. Figure 2 demonstrates the discourse complexity for five students with time. The results indicate that if a particular user role can be seen in consecutive posts the level of complexity increases/decreases with minimum change and when there is a role change (e.g. IS \rightarrow IG or IG \rightarrow IS or O \rightarrow IG) there is a dramatic change in discourse complexity.

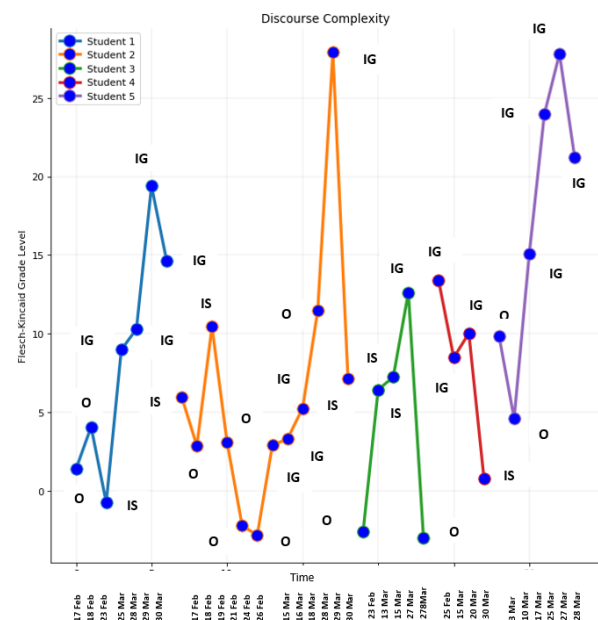


Figure 2: Discourse complexity across user roles with time

An initial exploratory analysis was performed on topic modelling using state of the art topic modelling technique known as Latent Dirichlet Allocation (LDA). We try to identify the topics that have been discussed in each user post and lecture transcripts. Figure 3 shows the percentage of each topic discussed by information givers and information seekers. According to the analysis, information givers are more involved in discussing the latter part of the course topics than information seekers while information seekers show interest towards the beginning of the lecture content. In future, further analysis will be performed to discover the reasons behind this observed trend.

Moreover, we calculated the affective state of each user posts using LIWC tool. Affect features measures the positive and negative sentiment and more specific emotion such as anger, anxiety and sadness. The results of one-way analysis of variance (ANOVA) show that information seekers express more lexical

³ <https://stanfordnlp.github.io/CoreNLP/>

semantics associated with affective state than information givers. Likewise, we would like to perform several other linguistic experiments to develop a linguistic framework that will demonstrate the linguistic characteristics of different student clusters in discussion forums.

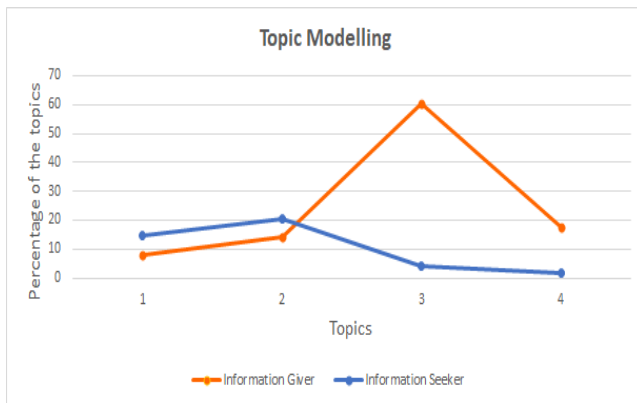


Figure 3: Topic modelling across user role

With the insight gained from existing experiments on user role and linguistic changes, our next step will be to predict the student grading using linguistic-only features. We have done a preliminary analysis on how to divide the student grading into different categories (e.g. pass, fail) and predicted student grades. However, further analysis needs to be performed on feature selection and deploying several other machine learning algorithms to fine-tune the obtained results.

5. CONCLUSION AND FUTURE WORK

The aim of our doctoral work is to understand student learning in MOOCs by investigating user roles and their associated linguistic change. As an initial stage, we have presented a multi-class user role classification in MOOC discussion forums using linguistic-only features with the intention of eliminating the drawbacks (e.g. contextual features) that exist in previous studies. Our model performed well compared to the baseline model, with 82.30 % of F-measure.

As future work, we try to integrate this classification with content and non-content user posts. Thus, it results in a novel classification on user role classification in MOOC discussion forum. On the other hand, our linguistic study gives us a clear differentiation of linguistics aspects associated with each role. Further, we hope to do a meticulous analysis to explore these patterns in future with the intention of discovering the possible reasoning behind the observed trends. Further analysis will be conducted to identify the discourse measures that can contribute to understanding student learning. The study would also like to explore diverse methods/techniques that can discover correlations between these linguistic measures and students' learning in MOOCs. Understanding how these linguistic measures can contribute directly/indirectly to students' learning will help us to propose novel methods to understand students' learning in an online learning environment. In addition, experiments will be performed to identify the correlations between the external factors (e.g. course structure, assignment deadlines) and user role transformations.

As a proof of concept, our technique demonstrated the potential of identifying the linguistic behaviours for each user role. This novel

approach holds a great promise for user role classification and the associated linguistic behaviour in MOOC discussion forums. Additionally, we believe that tracking these role changes and associated linguistic changes will help to understand the student learning in MOOC discussion forums. Thus, this doctoral work, will eventually try to find an answer to 'are students' really learning from MOOCs?'

6. ADVICE SOUGHT

For this doctoral consortium, the study would like advice regarding the following concerns mainly focusing on linguistic study:

1. Discuss language and discourse measures that can contribute to understanding student learning.
2. Discussion on possible reasoning behind the observed trends (e.g. the readability level of the information giver is low (i.e. discourse complexity is high) when compared to the information seeker and other user roles, the level of information embeddedness (number of clauses) is high within the information giver compared to the remaining classes).
3. Discussions on understanding the correlations between external factors (e.g. course structure, learners' demographic) and learner's role (e.g., information seeker, information giver) transformations.
4. Discussions on how existing learning frameworks (e.g. ICAP framework) associate with learner roles.

7. REFERENCES

- [1] Anderson, T., 2004. Towards a theory of online learning. *Theory and practice of online learning* 2, 109-119.
- [2] Arguello, J. and Shaffer, K., 2015. Predicting speech acts in MOOC forum posts. In *Ninth International AAAI Conference on Web and Social Media*.
- [3] Crossley, S.A., Greenfield, J., and McNamara, D.S., 2008. Assessing text readability using cognitively based indices. *Tesol Quarterly* 42, 3, 475-493.
- [4] Daphne Koller, Ng, A., Do, C., and Chen, Z., 2013. Retention and Intention in Massive Open Online Courses: In Depth. *Educ. Rev.* 48, 3, 62-63.
- [5] Dowell, N.M., Brooks, C., Kovanović, V., Joksimović, S., and Gašević, D., 2017. The changing patterns of MOOC discourse. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, 283-286.
- [6] Hecking, T., Chounta, I.-A., and Hoppe, H.U., 2016. Investigating social and semantic user roles in MOOC discussion forums. In *Proceedings of the sixth international conference on learning analytics & knowledge*, 198-207.
- [7] Lundberg, J., Castillo-Merino, D., and Dahmani, M., 2008. Do online students perform better than face-to-face students? Reflections and a short review of some empirical findings. *RUSC. Universities and Knowledge Society Journal* 5, 1, 35-44.
- [8] Nguyen, D. and Rosé, C.P., 2011. Language use as a reflection of socialization in online communities. In *Proceedings of the Workshop on Languages in Social Media Association for Computational Linguistics*, 76-85.
- [9] Searle, J.R., 1976. A classification of illocutionary acts. *Language in society* 5, 1, 1-23.

Building Test Recommender Systems for e-Learning Systems

Oana Maria Teodorescu
University of Craiova
Department of Computer Science
Craiova, Romania
teodorescu_oanamaria@yahoo.com

ABSTRACT

Modern e-Learning systems offer a wide variety of functionalities, from basic ones like accessing courses or online communication to more advanced ones like providing personalized feedback or assets. The learning environment can benefit from recommendations by providing students with tailored learning pathways, or assessment materials, thus ensuring the personalization and adaptation of the e-Learning platform to the student's individual needs. It can also allow for tracking and evaluation of the learner's progress, showing potential for improving the user experience for both students and professors. This research aims to investigate how a recommendation system can be used for building personalized tests in the context of education. The system's main goal is to improve the efficiency of the overall testing activity of learners by recommending questions relative to their knowledge level. It extracts input data based on past test results and uses learning analytics to provide a personal ranking of questions for each student based on their personal and their peers' experience with the studied concepts in a course. Contributions are foreseen on four different levels. First is the design and implementation of the recommendation algorithm. Second, raw data needs to be pre-processed by defining and extracting the features that can be used as input for the recommendation algorithm. Third, a post-processing step is needed for applying data analytics, rules and constraints to the resulted model in order to obtain proper recommendations. Last but not least, the presentation layer must be updated by providing a user interface for students and professors.

Keywords

e-learning; recommender system; user customization

1. INTRODUCTION

Most recommenders aim at providing recommendations to users based on their personal likes and dislikes. These systems use a specific type of information filtering technique

that attempt to recommend information items to the user. In an e-learning environment, both personal and collective information should be taken into account, as well as the links/relationships between the concepts covered in a chapter, as these could provide an insight into the level of knowledge of the student and uncover the missing gaps in the learning process a student is going through for each course.

Using a recommender system, in the context of e-assessments (i.e. online tests), enables the personalization and adaptation of the e-learning platform to the student's individual needs. The information it provides also allows for the tracking and evaluation of a student's progress by both learner and professor. By further analysing it, the information can provide a better understanding and structuring of the material which follows in subsequent chapters, thus providing a more logical chaining of concepts covered for a specific course.

Question recommendations in a test can provide a useful tool in the learning process of students for both the student (through tailored learning paths) and professor (by employing the means of defining and refining learning materials to a more logical and easy-to-understand chain of topics) with a direct impact in the application domain of e-Learning.

2. RELATED RESEARCH

Basic techniques for recommender systems (collaborative, content-based, knowledge-based, and demographic techniques) have known shortcomings such as the well known cold-start problem for collaborative and content-based systems (what to do with new users with few ratings) and the knowledge engineering bottleneck [7] in knowledge-based approaches, as Wikipedia states in [8].

According to an MIT tutorial for SVD (Singular Value Decomposition) [3], calculating the SVD for a matrix M (i.e. finding U and V such that $M = U \times \Sigma \times V$) reduces to finding the eigenvalues and eigenvectors of MM^T and $M^T M$. The eigenvectors of MM^T make up the columns of U , while the eigenvectors of $M^T M$ make up the columns of V . Also, the singular values in Σ are the square roots of eigenvalues from MM^T or $M^T M$. These singular values represent the diagonal entries of the Σ matrix and are arranged in descending order. They are always real numbers. If matrix M is a real matrix, then U and V will also be real.

Recommender systems for e-Learning platforms are based on

Oana Maria Teodorescu "Building Test Recommender Systems for e-Learning Systems" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 810 - 814

many approaches like web mining and information retrieval [4], recommender systems based on the context [13] or even using intelligent agents [14]. One interesting approach of using collaborative filtering in e-Learning systems [2] was to assign greater weights for users with higher knowledge than users with lower knowledge and the authors propose some new equations in the nucleus of the memory-based collaborative filtering. Another interesting paper, presenting clear results regarding recommender systems in smart e-Learning environments shows their approach [9] along with their encouraging results and their aim to extend the system for more faculties.

A concept map or conceptual diagram is a diagram that depicts suggested relationships between concepts, which are defined as “perceived regularities or patterns in events or objects, or records of events or objects, designated by a label” and are depicted as shapes in the diagram [10]. It is a graphical tool that instructional designers, engineers, technical writers, and others use to organize and structure knowledge [6].

3. RESEARCH QUESTIONS

Research questions that need to be addressed are primarily related to the area of recommender systems and selecting the proper recommendations in the context of e-Learning. The main focus is on the actors, which are both learners and professors using the system. The following questions arise:

Q1. How can a recommender system be efficiently used in the context of e-Learning?

A recommender system can give the student personalized tests, find learning gaps and suggest areas of improvement or concept revisions.

What data can a recommender system use as input?

Data from previous usage of the system by the student and his/her peers is required for a good output of the recommender system.

Q2. Are all types of recommender systems “recommended” in e-Learning?

There are majorly six types of recommender systems which work primarily in the Media and Entertainment industry: Collaborative, Content-based, Demographic based, Utility based, Knowledge based and Hybrid recommender system. Which type is better suited for the e-Learning system or how can these types be merged to obtain the most of the recommender for specific target (students of a specific group age, area of study etc.)? This is one of the questions that the research aims to answer.

Q3. How can the recommender system help the student in his/her learning process?

For a student, the recommender system can potentially help the student see his/her current level of knowledge, provide ways of revision and improvement, provide a general indication of the final results before a potential test and assess accumulated knowledge over time.

Q4. How can the recommender system help the professor in the learning process of his/her students?

For a professor, the recommender system can potentially show a student’s progress, point out the difficulties each student has at certain areas of a course and moments when he/she might need a tutor’s help, provide hints on how a student is expected to perform at a test/exam, provide insight on how well the students acquire new concepts based on past ones, identify out-of-order topics or missing information from the course materials.

4. PROPOSED CONTRIBUTIONS

4.1 Research Context

Classical on-line learning environments aim to create a support for learners to get their learning resources and take exams or to be evaluated by the professors; the next learning environments should be more personalized, analyzing each users’ needs and adapting the interface to their concerns and needs. If we consider a usual learning platform, we can say that the learning progress should be considered to be good by the professor for every student, but not all the students have the same needs, nor do they have the same performances at school. A tool that employs a recommender system can create intelligent interfaces capable to adapt to the users’ specific needs, to aggregate learning materials in order to provide the content necessary for the user at that moment and to create an order ranking over the learning materials and among students.

4.2 Research Activities

The main goal of this research proposal is to enhance the effectiveness of the e-Learning environments. In order to achieve this goal, three prerequisites need to be accomplished.

4.2.1 Prerequisites

P1. Analysis and formalization of recommender system’s usage in e-Learning An in-depth study should be conducted in order to assess the way recommender systems are currently used in the e Learning environment. Furthermore, an investigation on new ways to integrate them, along with other information retrieval algorithms for the definition and refinement of the input and output data of the recommended items, into existing e-Learning tools should be made.

P2. Adaptation and definition of data analysis pipelines for input data provided by e-Learning systems A data analysis must be performed on data available for processing in the e-Learning environment in order to filter out unnecessary data, fill in missing information and/or transform it into input data which can be then used by the recommender system. The format of the input data for the recommender system’s algorithm must be defined and updated as the internal processes of the algorithm itself are also updated.

P3. Application and validation of recommender system in student’s and professor’s activity on e-Learning system The aim of the recommender system is to prove itself useful in the context of an e-Learning environment. For this, the recommender system must be integrated in tools that have real application in e-Learning. Upon using and refining its process, a validation must be performed in order to ensure that it provides an increase in productivity of the learning process of the students and the objective evaluation of the professors. For this, certain evaluation methodologies or metrics

have to be devised for a reliable evaluation of the improvements and benefits the new system offers when using the recommender system.

4.2.2 Methodology

Figure 1 presents a generic workflow with 3 main layers and 4 general steps for designing, implementing and validating a recommender system in the context of e-Learning. The division of the system into four steps is inspired by [1] in which the authors propose a framework for an adaptive learning of MOOCs. We identify 3 actors that influence the system: learners (i.e.: students), professors and data analysts.

Data Representation Layer. From this layer, data needs to be gathered. Most systems use a database or log files to keep raw data about users and their activity. The data is taken from this layer and transformed as needed for the recommendation process. This is a layer responsible for providing data input for the Learning Analytics layer. The steps that manipulate data from this layer are the Data gathering and Pre-processing steps. The actors involved in this layer and the associated steps are the data analysts.

Learning Analytics Layer. This layer is responsible for data processing and data analytics, transforming data, defining rules and constraints, preparing data for the algorithms used and applying the final recommendation engine. It communicates with the Data Representation layer for data input and Presentation layer for data output. The specific steps for this layer are the Pre-processing step (for transforming data to the needed format and building data models) and Recommendation step (for employing needed rules, constraints, custom logic and algorithms needed for the recommendation engine). The actors involved in this layer are both data analysts (for building the data model, defining pre-processing logic, rules and constraints, defining and refining the recommendation engine based on experimental results) and users interested in the system (learners and professors) as they are the ones that enforce the domain-level constraints.

Presentation Layer. This layer is responsible with defining the graphical user interface of the system and providing services to the users, such as learner self-testing and online communication (a key feature in modern e-Learning platforms). It communicates with the Learning Analytics layer for data input. The Presentation step is specific to this layer and handles the user interface aspects of the end-user application. The actors involved in this layer are learners and professors which actively use the features of the system.

5. CURRENT STATUS

As research status, three papers have been written so far and an incremental approach is being used to actively improve the recommendation engine based on past results.

In paper [5], which I have co-authored, a custom recommender system based on SVD has been implemented in the context of extending an existing e-learning platform used for distance-education students enrolled in our local university. The recommender has been subsequently tested on students from our university in collaboration with professors for defining the pool of questions and concept maps in the system, with small adjustments being applied after

each year of study. The initial implementation of the recommender, presented in [5], was relying on a collaborative SVD algorithm applied on aggregated test results from all students enrolled in a course. The algorithm selected the proper questions from the available pool of questions, with the only constraints on question repetition and unknown question exploration in case of no questions in the initial aggregation matrix. This approach suffered from the cold-start problem, since it first resulted in generating random recommendation vectors and gradually getting to the desired recommendation mechanism.

Data visualization was implemented in the second paper [11] by building a concept map for the course and assigning concepts to each question in the system. This way, a concept map status could be generated for each student after each test, in which the concept would be colored in red/orange/green to highlight the progress of the student. Greener nodes would indicate that a student is starting to answer most questions in the concept correctly, while redder nodes would indicate that the student answered most questions in the concept incorrectly. Orange nodes were the middle point of the representation, signaling a half-correct distribution of the answers for that concept.

After the first experiment, a custom validation mechanism was implemented, as part of the third paper [12], in which a special function called Correctly Recommended Concepts (CRC) function was implemented. The CRC function was defined as a set of CRC values, plottable for each individual student. A concepts for revision functionality was implemented, which enabled the professor to mark which concepts should be recommended next based on a previous test result of a student. The CRC value was then computed for each test of a student (except the first, for which no revision concept would be defined) as the accuracy of matched concepts by the recommender system relative to the revision concepts, marked by the professor, on a scale of 0 to 1. More specifically, it was computed by dividing the number of matched concepts to the total number of concepts in the test. These values were then plotted for the student on a timeline having the test number on the X axis and CRC value on the Y axis. By observing the slope between tests, the relative performance could be computed by distinguishing 2 main categories of situations: *negative slope* for learners with a lower performance for the student in the next test and *zero or positive slope* for improvements in the performance of the student in the next test.

A second experiment has been conducted, with results to be analysed comparatively for students in a subsequent year of study as part of an incremental approach of improving the system based on results from previous work. The recommendation engine has been refined to consider the order defined in the concept map when choosing questions from the concepts. Also, the cold-start problem has been eliminated by using the old model alongside the new one, as previous test results from the former year have not been deleted from the database and will be used when aggregating input matrix data for the recommender system.

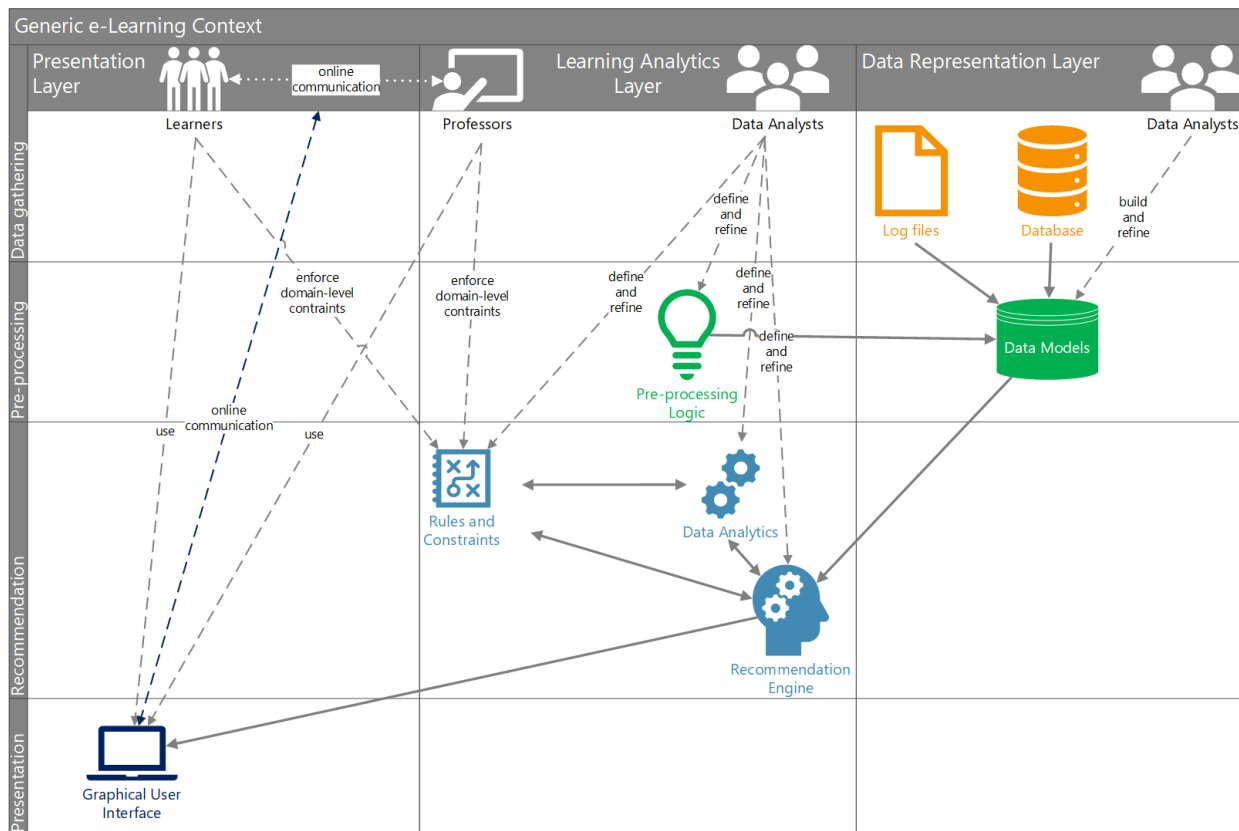


Figure 1: Generic layers and steps for defining a recommendation engine in the context of e-Learning

6. CONCLUSIONS

The main goal of the recommender system is to provide a personalized set of questions depending on both their current status and the status of their peers that have previously taken tests. Visual analytics of the experimental results in terms of knowledge coverage of the concept map show promising initial results.

Future work may regard not only the correctness by which the recommender manages to assign questions from right concepts, but also checking if recommended questions improve the student's learning rate or knowledge level. More work needs to be performed in terms of defining and integrating appropriate quantitative and qualitative metrics for measuring accumulated knowledge with and without the usage of the recommender system. Another future plan is providing the recommender as a software package such that integration into other e-Learning platforms can also be achieved.

7. REFERENCES

- [1] ARDCHIR, S., TALHAOU, M. A., AND AZZOUAZI, M. Towards an adaptive learning framework for moocs. In *MCETECH* (2017).
- [2] BOBADILLA, J., SERRADILLA, F., AND HERNANDO, A. Collaborative filtering adapted to recommender systems of e-learning. *Knowledge-Based Systems* 22, 4 (2009), 261 – 265. Artificial Intelligence (AI) in Blended Learning.
- [3] HOEKSTRA, R. The knowledge reengineering bottleneck. *Semant. Web* 1, 1,2 (Apr. 2010), 111–115.
- [4] KHRIBI, M. K., JEMNI, M., AND NASRAOUI, O. Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval. In *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on* (2008), IEEE, pp. 241–245.
- [5] MIHAESCU, C. M., TEODORESCU, O. M., POPESCU, P. S., AND MOCANU, M. L. Learning analytics solution for building personalized quiz sessions. In *2017 18th International Carpathian Control Conference (ICCC)* (2017), IEEE, pp. 140–145.
- [6] NOVAK, J. D., AND CAÑAS, A. J. The theory underlying concept maps and how to construct and use them.
- [7] RICCI, F., CAVADA, D., MIRZADEH, N., AND VENTURINI, A. Case-based travel recommendations. *Destination Recommendation Systems: Behavioural Foundations and Applications* (2006), 67.
- [8] SHANI, G., HECKERMAN, D., AND BRAFMAN, R. I. An mdp-based recommender system. *J. Mach. Learn. Res.* 6 (Dec. 2005), 1265–1295.
- [9] SOONTHORNPHISAJ, N., ROJSATTARAT, E., AND YIM-NGAM, S. Smart e-learning using recommender system. In *Proceedings of the 2006 International Conference on Intelligent Computing: Part II* (Berlin, Heidelberg, 2006), ICIC'06, Springer-Verlag,

- p. 518–523.
- [10] SOWA, J. F. Conceptual structures: information processing in mind and machine.
 - [11] TEODORESCU, O., POPESCU, S. P., AND MIHAESCU, M. C. Taking e-assessment quizzes-a case study with an svd based recommender system. In *Intelligent Data Engineering and Automated Learning – IDEAL 2018* (2018), Springer International Publishing, pp. 829–837.
 - [12] TEODORESCU, O., POPESCU, S. P., MOCANU, M., AND MIHAESCU, M. C. Custom validation procedure for tesys recommender system. In *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)* (2019), IEEE, pp. 1–6.
 - [13] VERBERT, K., MANOUSELIS, N., OCHOA, X., WOLPERS, M., DRACHSLER, H., BOSNIC, I., AND DUVAL, E. Context-aware recommender systems for learning: a survey and future challenges. *IEEE Transactions on Learning Technologies* 5, 4 (2012), 318–335.
 - [14] ZAÍANE, O. R. Building a recommender agent for e-learning systems. In *Computers in education, 2002. proceedings. international conference on* (2002), IEEE, pp. 55–59.

Crowd-sourcing and Automatic Generation of Semantic Information in Blended-Learning Environments

Elad Yacobson
Department of Science Teaching
Weizmann Institute of Science
Herzl St 234
Rehovot, Israel
elad.yacobson@weizmann.ac.il

ABSTRACT

Personalized learning environments rely on repositories of digital learning materials, and on meta-data that provides semantic information about the digital content. The semantic information is typically generated by domain experts, but this process is very time consuming, and fails to address the dynamic nature of the content and the contexts in which it is used. In addition, experts may fail to capture semantic properties that are not within their area of expertise. Overall, expert-based semantic generation processes do not scale, and produce limited information. Thus, the goal of my research is to study means to scale and improve the process of collecting and updating semantic information, using two different approaches: crowdsourcing from teachers and learners and automatic tagging that is based on machine-learning algorithms. As a proof-of-concept, two pilot experiments were conducted: the first was with two groups of physics teachers who are using an Open Educational Repository. The main goal was assessing the quality of the semantic information that the teacher-sourcing produces, and factors affecting it. The second experiment aimed at automatic tagging, and focused on comparing several ML approaches to automatically tag learning resources in a K-12 Math online learning environment. In this paper I will present the preliminary findings from these experiments, discuss future directions for my research, and seek advice concerning several issues involved with my research.

Keywords

Personalized Learning, Semantic Information, crowdsourcing

1. INTRODUCTION

Personalized learning environments rely on repositories of digital learning materials (e.g., interactive questions, online labs, videos), and on meta-data that provides rich semantic

information about the digital content. The term ‘semantic information’ refers to information describing the content and different attributes of the online learning resources, such as the topic, the level of difficulty, its intended use - whether as a test, class practice or homework, which grade it is appropriate for, the estimated amount of time required to complete the activity, the technological aids required for it (e.g., a computer, projector, mobile devices), and more.

The semantic information is fundamental to the ability of AI agents to make ‘intelligent’ decisions such as recommending content to learners, to assist teachers in search & discovery of learning resources, and for re-using and sharing materials between contexts [1, 2, 3]. However, while high-quality digital content is in many cases readily available on the web, it is the semantic information that is usually missing, inadequate, or partial. Thus, having scalable processes for generating high-quality semantic information can contribute significantly to the development of personalized learning environments.

Semantic information is typically generated by domain experts, but this process is very time consuming, and the experts may fail to capture semantic properties that are not within their area of expertise [5]. In addition, the content repository and the context in which it is used are dynamic, requiring frequent revisions and updates. Overall, expert-based semantic generation processes do not scale, and produce limited information. My research aims to address these issues, by studying means to produce semantic information at scale, as detailed in the next section.

2. RESEARCH DIRECTIONS

The high-level goal of my research is to study two main approaches for collecting and updating semantic information: The first is crowdsourcing (more accurately: teacher- and learner-sourcing, which are the terms that are used hereafter), and the second is automatic tagging using machine learning algorithms. More specifically, this goal is further divided into the following issues:

Semantic Information Required. The first issue that I want to examine is what types of semantic information assist teachers in search & discovery of educational resources in open repositories. With the transfer of a growing number of

Elad Yacobson "Crowd-sourcing and Automatic Generation of Semantic Information in Blended-Learning Environments" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 815 - 817

teachers to blended learning, teachers often rely on learning resources found online. These resources are commonly organized in Open Educational Repositories (OER) that offer teachers a large pool of instructional materials. For teachers, selecting appropriate and effective instructional materials from an OER is a challenging and time-consuming process. Thus, I wish to explore what semantic information is required for supporting teachers in this process.

Sources for Obtaining Semantic Information. As aforementioned, I will focus on two approaches for obtaining semantic information about the learning resources: crowdsourcing from teachers and learners, and automatic tagging that is based on ML algorithms. I will compare two aspects of the information obtained from these approaches: the accuracy of the information, and coverage – how much information can be obtained from each of these sources.

Factors That Affect the Quality of Teacher and Learner-Sourced Information. The goal of this research direction is to study the different factors that affect the quality of teachers and learner-sourced semantic information. I will address both the issue of teachers' and learners' *ability* to accurately tag learning resources with semantic information, and their *motivation* to do so. In terms of ability, I will study issues and task definitions that support accurate tagging. For instance, can teachers and students tag resources without having full knowledge of the taxonomy from which the tags are taken? to which resolution do teachers and learners need to go in analyzing the questions in order to provide accurate tags? The second issue is teachers and learners motivation to contribute time and effort to tagging, as this is a time-consuming and cognitively demanding process, with no perceived reward. Providing incentives for crowdsourcing is a known issue [4], and it is reasonable to assume that engaging teachers and learners in crowdsourcing would require appropriate incentive design.

Effect of Tagging Process on Teachers and Learners. The last aspect of crowdsourcing that I wish to examine is the effect (if there is any) of the tagging process on teachers' professional development, and on students' learning. With respect to teachers, I will focus on the effect of participation on their ability to provide personalized learning and adapt tasks to individual needs of different students. With respect to learners, I intend to focus on whether the reflective nature of the tagging process contributes to student understanding, as reflective processes has been repeatedly shown to improve learning.

3. PRELIMINARY RESULTS

To date, two pilot experiments were conducted, each addressing a different approach for obtaining semantic information. The first experiment was held with two groups of Physics teachers. The teachers were requested to tag questions taken from a blended-learning environment named *PeTeL* (described below), and their tagging was compared to that of domain experts. In the second experiment, a supervised machine-learning approach was applied to ~ 400

activities taken from a Math learning environment named *STEP* (see below), which are tagged according to different dimensions, such as their topic (Geometry, Algebra, Verbal Problems, Infinitesimal Calculus etc.)

3.1 First Experiment - Teacher Sourcing

The first experiment was designed as a proof-of-concept for teachers' ability to accurately tag learning resources with semantic information. The participating teachers were requested to tag questions taken from a learning unit on Magnetism according to a detailed taxonomy prepared by a group of expert teachers and researchers.

Learning Environment - PeTeL. The experimental setup is based on a learning environment named PeTeL, which is both an OER, and an LMS that also includes social network features and learning analytics tools. It is developed within the Department of Science Teaching at Weizmann Institute of Science, with the goal of providing STEM teachers with a blended learning environment for personalized instruction. PeTeL is divided into separate modules for each subject matter: Biology, Chemistry and Physics. The Physics module is currently being used by approximately 200 teachers and 7000 high school students. All the teachers who participated in the experiment use PeTeL in their classes.

Procedure and Results. Two groups of Physics teachers participated in this experiment. The first group consisted of eight teachers who were presented each with three questions from PeTeL. Each question contains a picture or diagram of a certain Physics situation (e.g. a particle moving through a magnetic field, or an electric circuit), and a question regarding that diagram (See example in Figure 1). For each question i , the teachers were presented with four tags. Then, for each tag, the teachers were requested to decide whether it applies to i . Overall, we received 95 responses. In 74 out of 95 responses, the teachers agreed with the domain expert as to whether the content knowledge described in the tag is required for solving the question (78% agreement, Cohen's kappa: 0.56).

A rectangular frame with sides a and b , is shown in the following diagram. An electric current is flowing through each of the frame's sides in counter-clockwise direction. The frame is located in a magnetic field entering the page's plain, as shown in the diagram.

- * what is the direction of the magnetic force working on side a of the frame?
- * what is the direction of the magnetic force working on side b of the frame?
- * what is the direction of the magnetic force working on side c of the frame?
- * what is the direction of the magnetic force working on side d of the frame?

Does solving this question require the following concepts?

- * The magnetic field creates a force over a current-carrying wire - yes / no
- * Effect of different parameters on the force: magnitude of magnetic field and of current - yes / no
- * The magnetic force's direction is vertical to the direction of the magnetic field and to the direction of electric charges - yes / no
- * Effect of different parameters on the force: the angle between the direction of the field and the direction of the current - yes / no

Figure 1: Tagging Task Example

The second part of the experiment took place about a month after the first one, with a different group of seven Physics teachers, and followed a similar protocol. A total of 56 responses were collected. In 43 out of 56 responses, the teachers agreed with the domain expert as to whether the content knowledge described in the tag was required for solving the questions (77% agreement, Cohen's kappa: 0.54).

3.2 Second Experiment - Automatic Tagging

The second experiment was conducted in the context of STEP, an OER for junior-high and high-school Math, which was developed by the Department of Math Education in the University of Haifa.

Procedure. 407 learning activities were taken from STEP. Fifty keywords were selected as features (e.g., 'angle', 'function', 'speed', 'derivative', 'linear', 'sinus', etc.). Each activity was encoded as a one-hot vector according to the presence of these keywords, and labeled with its Math topic (Geometry, Algebra, Verbal Problems, etc.). Then, three ML algorithms (Naive Bayes, Random Forest, and Logistic Regression) were applied to the data in an attempt to evaluate the feasibility of classifying activities into topics based on these features.

Results. Measured with k-fold cross-validation, the accuracy of the classification produced by the ML algorithms was 95% (achieved by the Naive Bayes and the Random Forest algorithms).

3.3 Conclusions

The results of these two small-scale experiments suggest that regarding teacher-sourcing, when the tagging task is formulated in a certain way (e.g., "yes/no" questions), teachers can tag items relatively accurately (Cohen's kappa: 0.56) without being trained on the taxonomy from which the tags are taken. Regarding automatic tagging by ML algorithms, these preliminary results are encouraging as to the ability to produce quality semantic information without the need for human intervention.

On the next step, we intend to run these experiment on a larger scale, using a technological tool to teacher-source semantic information from a much larger pool of teachers, to expand our work to learner-sourcing as well, and to apply learning algorithms to a multitude of learning resources in an attempt to reach much more fine-grained semantic information.

4. PROPOSED CONTRIBUTION

I hope that my work will have both a practical contribution to the learning environments that I study, and through this, to teaching and learning, and will contribute to EDM research by providing a better understanding of effective means to enrich learning environments with semantic information.

5. DISCUSSION AND ADVICE SOUGHT

I seek advice regarding four major issues in my research: the first is *what are the most effective means to enhance teachers' and learners' motivation to invest time and effort*

in the tagging process? In this regard, since we saw indications that teachers' motivation affects the quality of their tagging, I feel that positive incentives, rather than negative ones (e.g., requiring participation for receiving access to materials) are more likely to produce quality results.

The second issue is *how to optimize the relationship between coverage (i.e. how many tags are requested from each teacher or learner) and motivation.* On one hand, presenting the teacher/learner with numerous requests for tagging could easily deter him/her and would result in low rates of participation. On the other hand, minimizing the interaction with the user would result in low coverage.

The third aspect is how to evaluate the quality of the semantic information received from teachers and learners? After receiving tags produced by either teachers or learners, there is the question of how reliable those tags are. Possible solutions are random evaluation by an expert, or wisdom-of-the-crowd based ranking solutions.

And last, regarding the process of automatic tagging – a main challenge is abstracting different types of information representation (text, figures, symbols) into a common layer of semantic meaning, probably relying on NLP, object recognition, etc. I would appreciate receiving information regarding relevant research that I can build upon.

6. REFERENCES

- [1] T. Anderson and D. Whitelock. The educational semantic web: Visioning and practicing the future of education. *Journal of interactive Media in Education*, 2004.
- [2] L. Aroyo and D. Dicheva. The new challenges for e-learning: The educational semantic web. *Journal of Educational Technology & Society*, 7(4):59–69, 2004.
- [3] I. I. Bittencourt, S. Isotani, E. Costa, and R. Mizoguchi. Research directions on semantic web and education. *Interdisciplinary Studies in Computer Science*, 19(1):60–67, 2008.
- [4] N. T. Heffernan, K. S. Ostrow, K. Kelly, D. Selent, E. G. Van Inwegen, X. Xiong, and J. J. Williams. The future of adaptive learning: Does the crowd hold the key? *International Journal of Artificial Intelligence in Education*, 26(2):615–644, 2016.
- [5] G. McCalla. The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learners. *Journal of Interactive Media in Education*, 2004(1), 2004.

The Learner Data Institute: Big Data, Research Challenges, & Science Convergence in Educational Data Science

EDM 2020 Workshop Description

Vasile Rus

University of Memphis

vrus@memphis.edu

Stephen E. Fancsali

Carnegie Learning, Inc.

sfancsali@carnegielearning.com

ABSTRACT

We describe a half-day online workshop for researchers interested in learning about—and contributing to—the work of the Learner Data Institute (LDI), an initiative funded by the U.S. National Science Foundation (NSF) and based at the Institute for Intelligent Systems, University of Memphis, in collaboration with Carnegie Learning, Inc., and other partners. LDI's mission is to foster and support science convergence to address major challenges in learning with technology (online and blended) through an expanding, global network of expert panels, school-based practitioners, interdisciplinary research teams, and task-oriented special interest groups. Founding members of the LDI have been working together to plan mechanisms and processes for building shared understanding, and for exploring methods of extracting and packaging actionable knowledge from the massive datasets generated by current computer-supported instructional systems and from the stores of text, sound, graphics, video, and other data modalities that are rapidly accumulating on cloud-based servers around the world. The community at large is invited to participate in the workshop to hear about our work, help us refine our ideas through paper and talk contributions, and explore how they too can become involved in this important new enterprise.

Keywords

big data in education, educational data science, educational data mining, learning analytics, educational technology, science convergence, interdisciplinary research, trans-disciplinary research, multi-disciplinary research.

1. WORKSHOP DESCRIPTION

1.1 Interdisciplinary solutions

Historically, the proceedings of the *International Conference on Educational Data Mining* (and related conferences, including the *International Conference on Artificial Intelligence in Education*, the *ACM Conference on Learning at Scale*, and *Learning Analytics and Knowledge*) demonstrate inherent linkages across traditional and emerging academic disciplines and research areas. Vasile Rus and Stephen Fancsali "The Learner Data Institute: Big Data, Research Challenges, and Science Convergence in Educational Data Science." In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*. Anna N. Rafferty, Jacob Whitehill, Violetta Cavallari-Storza, and Cristobal Romero (eds.) 2020, pp. 818 - 820

as, but not limited to, psychology, economics, cognitive and learning science(s), mathematics, computer science (e.g., machine learning, artificial intelligence), statistics, human-computer interaction, public policy, education, neuroscience, social work, moral and political philosophy, and any of a number of sub-fields and research areas at the intersection of these disciplines.

The need for such multi/inter/trans-disciplinary solutions is even more relevant today as the vast and diverse repositories of digital data available can make such solutions viable. Indeed, recognizing both substantial scientific challenges and the need for innovative scientific frameworks to solve them, the U.S. National Science Foundation has identified the notion of “convergence” research as one of ten “big ideas” for its on-going investment strategy [1]. Two attributes are crucial to NSF’s notion of convergence research [2], namely that such research is “driven by a specific and compelling problem” and emphasizes “deep integration across disciplines.” Such integration is achieved when:

“...experts from different disciplines pursue common research challenges, [and] their knowledge, theories, methods, data, research communities and languages become increasingly intermingled or integrated. New frameworks, paradigms or even disciplines can form sustained interactions across multiple communities” [2].

The NSF-funded Learner Data Institute (LDI) focuses on such science convergence solutions for major challenges in learning with technology (online and blended). Furthermore, LDI will contribute to at least two of the ten new Big Ideas for Future Investment announced by NSF: Harnessing the Data Revolution for 21st Century Science and Engineering (HDR) and The Future of Work at the Human-Technology Frontier (FW-HTF).

1.2 The Learner Data Institute

LDI is an NSF-funded “data-intensive research in science and engineering” (DIRSE) initiative seeking to set out compelling, specific, big data research challenges for educational data science researchers and large-scale scientific and data convergence approaches to address them.

The LDI will help us learn: (1) how to transform a far-flung group of interdisciplinary researchers, developers, and practitioners into

a community of practice that can fully exploit the data revolution through data and science convergence; (2) how adaptive instructional systems and data science can be used as research vehicles to further our understanding of how learners learn; (3) to explore the human-technology partnership with data and data science to improve learners' and teachers' ability to employ technology in a way that facilitates learning, while at the same time improving the affordability, effectiveness, scalability of these systems; and (4) more generally, how to extend the frontiers of data science to include: new methods of data collection and design; more interpretable machine learning methods (e.g., by combining deep learning with more interpretable inference frameworks like Markov Logic); scalable new algorithms (e.g., for joint inference in Markov Logic Networks); and methods for identifying causal mechanisms from unstructured, semi-structured, and structured data.

More specifically, LDI contributors from university-based research groups, industry, and government are focusing on cutting-edge, big data approaches to *assessment, learner modeling, instructional design, modeling subject-area domains in instructionally useful ways, socio-cultural aspects of learning, ethical aspects of working with learner data, and the human-technology frontier*, among other areas of interest.

Approximately 40 LDI contributors were recently surveyed as a part of the initiative to envision the future of convergence research in educational data science; they were asked to identify areas of challenge and societal need with respect to improving education and learning, compelling opportunities, and ways in which big data can be harnessed to address both. Portions of this workshop will be devoted to presenting and discussing the findings of this survey. In addition, the workshop convenes a diverse set of researchers and developers, some associated with LDI and others not, working with big data from contexts in which learning takes place, seeking to better understand state-of-the-art interdisciplinary research as well as compelling, specific societal needs and challenges and the scientific, big data frameworks that might be leveraged to solve them in the future.

We expect that this convening of a diverse, highly experienced group of researchers will stimulate substantial growth and interest in the notion of science convergence, including helping to set the direction of the LDI and the framework that it is tasked by NSF with developing.

2. WORKSHOP FORMAT

The half-day workshop will take the form of an introductory talk introducing the LDI, presenting results of the LDI contributor survey, and situating those results within the goals of LDI and the broader notion of convergence research for educational data science. Two invited speakers will deliver keynote talks (including Q&A) laying out their visions for convergence research in educational data science, situating this idea within their individual research program(s), and/or discussing the results of the LDI survey. A selection of peer-reviewed contributed research papers (generally concerned with state-of-the-art big data methodology, applications, and research in educational data science, ideally with an emphasis on science convergence) and/or position papers (on similar topics with an eye toward where future research *should* be directed) will also be presented as a part of the workshop. A panel discussion will also take place, involving a moderator, keynote speakers, and 1-3 invited contributors.

3. SCHEDULE

The half-day workshop will include introductory remarks presenting an overview of the LDI as well as its recent survey/envisioning results, two invited keynote talks, and several presentations from selected paper submissions and/or invited contributors from LDI and the broader community at large. Time will be allotted for Q&A for each presentation as well as a panel discussion given sufficient time.

4. SPEAKERS

4.1 Overview of the LDI Mission, Activities, & Accomplishments: Vasile Rus, University of Memphis, Department of Computer Science & Institute for Intelligent Systems

Vasile Rus, William Dunavant Professor of Computer Science and Institute for Intelligent Systems, University of Memphis, will provide the introductory presentation and is Lead Principal Investigator of the Learner Data Institute. His research, funded by NSF, the U.S. Department of Education Institute for Education Sciences, the Office of Naval Research, and other funding agencies, centers on topics of natural language processing and understanding, including semantic similarity, question answering, and knowledge representation, especially with applications to adaptive instructional systems and software defect knowledge management. Before joining the University of Memphis, Dr. Rus received a Ph.D. in Computer Science at Southern Methodist University and was an Assistant Professor of Computer Science at Indiana University.

4.2 LDI – The Developer and Practitioner Perspective: Stephen E. Fancsali, Carnegie Learning, Inc.

Stephen E. Fancsali is Director of Advanced Analytics at Carnegie Learning, Inc., and Co-PI of the Learner Data Institute. His work focuses on statistical and causal modeling using data from adaptive systems for learning like Carnegie Learning's *MATHia* intelligent tutoring system (formerly *Cognitive Tutor*). This work includes developing practical progress monitoring metrics usable by teachers and learners in K-12 classrooms, statistical early warning systems that indicate when students are struggling unproductively, causal modeling of learner behavior, and scaling up semi-automated methods for improving cognitive models that underlie such adaptive learning systems to produce deployable instructional improvements. Before joining Carnegie Learning, Inc., he received a Ph.D. in Logic, Computation, & Methodology at Carnegie Mellon University.

4.3 Invited Keynote Speaker: Jason Hartline, Northwestern University, Department of Computer Science

4.3.1 Keynote Abstract

The talk will provide an overview of a peer grading system that is under development at Northwestern U. In courses that use the system it has (a) reduced the grading load of course staff by over 75%, (b) expanded and improved the students' interaction with the course material, and (c) improved turn-around time of feedback on student work (students receive comments on their work after three days, rather than two weeks). As a research platform, this system enables a dialogue between theory and practice for algorithms, machine learning, data science, and mechanism design. Of particular focus for the talk is on

mechanisms that incentivize peers to produce accurate reviews and the connection between these mechanisms and auction theory.

4.3.2 Keynote Speaker Biography

Jason Hartline is a professor of computer science at Northwestern University and a co-director of the Institute for Data, Econometrics, Algorithms, and Learning (IDEAL). His research introduces design and analysis methodologies from computer science to understand and improve outcomes of economic systems. Optimal behavior and outcomes in complex environments are complex and, therefore, should not be expected; instead, the theory of approximation can show that simple and natural behaviors are approximately optimal in complex environments. This approach is applied to auction theory and mechanism design in his graduate textbook *Mechanism Design and Approximation*, which is under preparation (<http://jasonhartline.com/MDnA/>). Professor Hartline received his Ph.D. in 2003 from the University of Washington under the supervision of Anna Karlin. He was a postdoctoral fellow at Carnegie Mellon University under the supervision of Avrim Blum and subsequently a researcher at Microsoft Research in Silicon Valley. He joined Northwestern University in 2008.

4.4 Invited Keynote Speaker: Carolyn Penstein Rosé, Carnegie Mellon University, Language Technologies Institute & Human-Computer Interaction Institute – “Towards Computer-Supported Collaborative Learning in the Workplace Enabled by Language Technologies”

4.4.1 Keynote Abstract

Well meaning companies offer training opportunities to their employees, but when push comes to shove, companies are known to push for short-term productivity over learning and higher productivity in the long term. The practical goal of the research is to enable learning during work, with a focus on software development teams.

Building on over a decade of AI-enabled collaborative learning experiences in the classroom and online, in this talk we report our work in progress beginning with classroom studies in large online software courses with substantial teamwork components. Project courses provide an effective test bed to begin our investigations due to similar tensions imposed by the reward structure. Project courses are believed to be valuable experiences for students to engage in reflection on concepts while applying them in practice. However there is a concern that the reward structure encourages students to engage in performance oriented behaviors, such as the most capable student taking on the lion's share of the work while leaving the others behind. These behaviors undercut the opportunity to use the project experience for each student to gain practice and for the students to reflect together on underlying concepts. In our classroom work, we have adapted an industry

standard team practice referred to as Mob Programming into a paradigm called Online Mob Programming (OMP) for the purpose of encouraging teams to reflect on concepts and share work in the midst of their project experience. At the core of this work are process mining technologies that enable real time monitoring and just-in-time support for learning during productive work. This talk will offer an overview of a series of classroom studies and introduce a corpus available through Learnsphere.org's DiscourseDB facilities:

<https://erebor.lti.cs.cmu.edu/discoursedb/index.html>.

4.4.2 Keynote Speaker Biography

Dr. Carolyn Rosé is a Professor of Language Technologies and Human-Computer Interaction in the School of Computer Science at Carnegie Mellon University. Her research program is focused on better understanding the social and pragmatic nature of conversation, and using this understanding to build computational systems that can improve the efficacy of conversation between people, and between people and computers. In order to pursue these goals, she invokes approaches from computational discourse analysis and text mining, conversational agents, and computer supported collaborative learning. Her research group's highly interdisciplinary work, published in over 240 peer reviewed publications, is represented in the top venues in 5 fields: namely, Language Technologies, Learning Sciences, Cognitive Science, Educational Technology, and Human-Computer Interaction, with awards in 3 of these fields. She is a Past President and Inaugural Fellow of the International Society of the Learning Sciences, Senior member of IEEE, Founding Chair of the International Alliance to Advance Learning in the Digital Era, and Co-Editor-in-Chief of the *International Journal of Computer-Supported Collaborative Learning*. She is a 2020-2021 AAAS Fellow under the Leshner Institute for Public Engagement with Science, with a focus on public engagement with Artificial Intelligence.

5. WORKSHOP WEB SITE

<https://sites.google.com/view/learnerdatainstitute/ldiedm>

6. ACKNOWLEDGMENTS

The Learner Data Institute is supported by the U.S. National Science Foundation under DRK-12/DIRSE Award #1934745. All opinions and findings stated or implied are solely those of the authors.

7. REFERENCES

- [1] U.S. National Science Foundation. 2017. NSF's 10 big ideas. https://www.nsf.gov/news/special_reports/big_ideas/index.jsp Last accessed 9 January 2020.
- [2] U.S. National Science Foundation. n.d. Convergence research at NSF. <https://www.nsf.gov/od/oia/convergence/index.jsp> Last accessed 9 January 2020

An Introduction to Neural Networks

Agathe Merceron
Beuth University of Applied Sciences Berlin
merceron@beuth-hochschule.de

Ange Tato
Université du Québec à Montréal
nyamen_tato.ange_adrienne@uqam.ca

ABSTRACT

In this tutorial, participants explore the fundamentals of feedforward neural networks such as the backpropagation mechanism and Long Short Term Memory neural networks. The tutorial also covers the basis of Deep Knowledge Tracing, the attention mechanism and the application of neural networks in education. There will be some hands-on applications on open educational datasets. The participants should leave the tutorial with the ability to use neural networks in their research.

A laptop capable of installing and running RapidMiner, Python and the Keras library is required for full participation in this tutorial.

Keywords

Neurons, Neural networks, LSTM, Attention mechanism.

1. INTRODUCTION

Neural networks (NN) are as old as the relatively young history of computer science: McCulloch and Pitts already proposed nets of abstract neurons in 1943 as Haigh and Priestley report in [5]. However, their successful use, especially under the form of convolutional neural networks (CNN) or Long Short Term Memory (LSTM) neural networks, in areas such as image recognition and language translation in the last years have made them widely known, also in the Educational Data Mining (EDM) community. This is reflected in the contributions that are published each year in the proceedings of the conference.

The upper green curve labeled “Neural Networks + LSTM” of Figure 1 shows the percentage of contributions (long and short papers, posters & demos, young research track, doctoral consortium, and papers of the industry track) that have used some kind of neural networks in their research while Table 2 -at the end of this paper- shows in the column “Total” the number of these contributions. Contributions that mention neural networks in the related works or future works only are not counted. One notices two jumps: in 2016 and 2019; the total goes from two to ten and then from 16 to 32 while the number of contributions goes from 147 to 132 and then from 112 to 139. This shows that neural networks are becoming more and more important in our field. In Figure 1, the blue curve “Neural Networks + no LSTM” gives the percentage of the contributions that have used neural networks other than LSTM neural networks, simply called LSTM in the following, while the orange curve “LSTM” shows the percentage of papers that have used LSTM in their research (these contributions might have used LSTM and also other kinds of neural networks). Till 2015, the green curve and the blue curve overlap, as there is no contribution using LSTM. In Table 2, the International Conference on Educational Data Mining (EDM 2020), Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 821 - 823

columns “Neural Network” and “LSTM” give the numbers instead of the percentages.

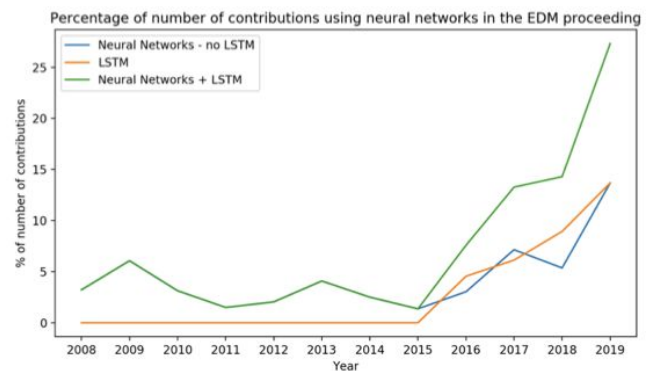


Figure1. Evolution over the years of the percentage of contributions using neural networks in the EDM proceedings.

Recognizing the growing importance of neural networks in the EDM community (see Figure 1 and Table 2), this tutorial aims to provide 1) an introduction to neural networks in general and to LSTM neural networks with a focus on the attention mechanism, and 2) a discussion venue on these exciting techniques. This tutorial targets 1) participants who have no or very little prior knowledge about neural networks and would like to use them in their future work or would like to better understand the work of others, and 2) participants interested in exchanging and discussing their experience with the use of neural networks.

A simple kind of neural network is a feedforward neural network also often called a multilayer perceptron. It propagates the calculation of each neuron from its inputs through all layers in a directed way forward to its outputs. In education, such NN are often used to predict the performance of students. The work of Romero et al. [12] presented at the first EDM conference in 2008 use them to predict the final mark of students in a course taught with the support of the learning platform Moodle.

While their primary use was in Natural Language Processing (NLP) Tasks, LSTM have recently been used in education and have achieved remarkable results [17, 16]. Opposed to feedforward neural networks that cannot remember the past, LSTM have cycles and are a kind of recurrent neural network. The LSTM [6] architecture can learn long-term dependencies using a memory cell that can preserve states over long periods. It is suitable for contexts where sequential information and temporal prediction is important such as in education, where we are interested in the prediction of students' outcome based on past

behavior. Deep Knowledge Tracing [8] is probably the best example of using LSTM to track students' knowledge states while they interact with a tutoring system. Numerous variants of LSTM have been proposed, such as the Gated Recurrent Unit (GRU) [3], or the LSTMs with Attention.

Attention [2] in machine learning refers to a model's ability to focus on specific elements in data. It helps the LSTM to learn where to look in the data. It was initially designed in the context of Neural Machine Translation using sequence to sequence (Seq2Seq or encoder-decoder) [13] models. However, since the attention mechanism can improve prediction results of NN models, it is now widely used in text mining in general. Especially in the education domain, it has been used for questions answering tasks, sequential modeling for student performance prediction or to predict essay or short answer scoring [18, 11].

2. THE TUTORIAL

2.1 Schedule

Table 1. Tutorial schedule

Time	Item
45 minutes	Introduction - Feedforward neural networks and backpropagation
45 minutes	Application - Discussion - Hands-on with RapidMiner
30 minutes	Break
60 minutes	LSTM and Attention Mechanism
60 minutes	Application - Implementation of a LSTM for student performance prediction - Discussion

2.2 Introduction to feedforward neural networks

This part begins with artificial neurons and their structure - inputs, weight, output, and the activation function - and the calculations that are feasible and not feasible with one neuron only. It continues with feedforward neural networks or multi-layer perceptrons (MLP). A hands-on example taken from [7] illustrates how a feedforward neural network calculates its output.

Further, this part introduces the backpropagation algorithms and makes clear what a feedforward neural network learns. Backpropagation is demonstrated with the hands-on example introduced before.

2.3 Application

This part discusses the use of feedforward neural networks in EDM research. These networks are often used to predict students' performance and students at-risk of dropping out, see for example [4, 1, 15]. However, other uses emerge. For example, Ren et al. use them to model the influence on the grade of course taken by a student of all other courses that the student has co-taken [10].

The main activity of this part is for participants to create, inspect, and evaluate a feedforward neural network with the free version of the tool RapidMiner Studio [9]. The data that will be used

comes from a German university. The task is to predict whether a student will drop out of a degree program. RapidMiner Studio is a graphical tool for Data Science which requires no programming. The tool will be introduced and participants will learn to load data, explore them, and classify them with neural networks. In particular, the following steps will be covered: discovering the operators "Neural Net" and "AutoMLP", cross-validation, models comparison, and grid optimization of the parameters with RapidMiner. Processes will be provided so that participants do not have to design them from scratch and can learn more efficiently.

2.4 LSTM

In this part of the tutorial, basic concepts of LSTM are covered. We will focus on how the different elements (cell, state, etc.) of the architecture work. Participants will learn how to use an LSTM for the prediction of learners' outcomes in an educational system. Concepts such as the Deep Knowledge Tracing (DKT) will be also covered.

2.5 Attention Mechanism

In this part, the attention mechanism is introduced. Participants will learn how this mechanism works and how to use it in different cases. We will explore concepts such as global and local attention in neural networks.

2.6 Application

In this hands-on part, we will explore existing real-life applications of LSTM (especially Deep Knowledge Tracing) in education. We will also explore the combination of LSTM with Expert Knowledge (using the attention mechanism) for Predicting Socio-Moral Reasoning skills [14]. Participants will implement an LSTM with an attention mechanism for the prediction of students' performance in a tutoring system. We will use Python especially the Keras library for coding. We will also use open educational datasets (e.g. Assisments benchmark dataset).

3. OBJECTIVES AND OUTCOMES

The objectives of this tutorial are twofold: 1) introduce the fundamental concepts and algorithms of neural networks to newcomers, and then build on these fundamentals to give them some understanding of LSTM and the attention mechanism; 2) provide a place to discuss and exchange about experiences while using neural networks with educational data.

Newcomers should leave the tutorial with a good understanding of neural networks and the ability to use them in their own research or to appreciate better research works that use neural networks. Participants already knowledgeable about neural networks get a chance to discuss and share about this topic and connect with others.

A website will be created to display important information to participants: schedule, slides, data, software to download and install.

Table 2. Number of contributions using neural networks in the EDM proceedings

Year	Neural Network	LSTM	Total	Number Contributions
2008	1	0	1	31

2009	2	0	2	33
2010	2	0	2	64
2011	1	0	1	67
2012	1	0	1	49
2013	4	0	4	98
2014	3	0	3	120
2015	2	0	2	147
2016	4	6	10	132
2017	7	6	13	98
2018	6	10	16	112
2019	19	19	38	139

4. REFERENCES

- [1] Berens, J., Schneider, K., Görtz, S., Oster, S., & Burghoff, J. 2019. Early Detection of Students at Risk—Predicting Student Dropouts Using Administrative Student Data from German Universities and Machine Learning Methods. *JEDM | Journal of Educational Data Mining*, 11(3), 1–41. <https://doi.org/10.5281/zenodo.3594771>
- [2] Chorowski, Jan K., Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, 2015. "Attention-based models for speech recognition." *In Advances in neural information processing systems*. pp. 577-585.
- [3] Chung, J., Gulcehre, C., Cho, K. and Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- [4] Dekker, G.V., M. Pechenizkiy, M. and Vleeshouwers, J.M. 2012. Predicting Students Drop Out: A Case Study. *In Proceedings of the 2nd International Conference on Educational Data Mining* (Cordoba, Spain, July 1-3). EDM'09, 41-50.
- [5] Haigh, T., Priestley, M.. 2020. Von Neumann Thought Turing's Universal Machine was 'Simple and Neat.': But That Didn't Tell Him How to Design a Computer. *Communications of the ACM*. 60, 1 (Jan. 2020), 26-32.
- [6] Hochreiter, Sepp, and Schmidhuber, J. 1997. "Long short-term memory." *Neural computation* 9.8: 1735-1780.
- [7] Mazur, M. A Step by Step Backpropagation Example. <https://mattmazur.com/2015/03/>
- [8] Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. 2015. Deep knowledge tracing. *In Advances in neural information processing systems*. pp. 505-513.
- [9] RapidMiner <https://rapidminer.com/products/studio/>
- [10] Ren, Z., , Ning, X., Lan, A.S., Rangwala, H. 2019. Grade Prediction Based on Cumulative Knowledge and Co-taken Courses. *Proceedings of the 12th International Conference of Educational Data Mining* (Montréal, Québec, Canada, July 2-5, 2019) 158-167.
- [11] Riordan, B., Horbach, A., Cahill, A., Zesch, T., & Lee, C. 2017. Investigating neural architectures for short answer scoring. *In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. pp. 159-168.
- [12] Romero, C., Ventura, S., Espejo, P.G, and Hervás, C. 2008. Data Mining Algorithms to Classify Students. *Proceedings of the 1st International Conference of Educational Data Mining* (Montréal, Québec, Canada, June 20-21, 2008) 8-17.
- [13] Sutskever, I., Vinyals, O., & Le, Q. V. 2014. Sequence to sequence learning with neural networks. *In Advances in neural information processing systems*. pp. 3104-3112.
- [14] Tato, A., Nkambou, R. and Dufresne, A. 2019. Hybrid Deep Neural Networks to Predict Socio-Moral Reasoning skills. *Proceedings of the 12th International Conference on Educational Data Mining (EDM'19)*. pp. 623-626
- [15] Wagner, K., Merceron, A., & Sauer, P. 2020. Accuracy of a cross-program model for dropout prediction in higher education. *In Workshop Addressing Drop-Out Rates in Higher Education ADORE'2020*, co-located with the 10th International Learning Analytics and Knowledge Conference, Frankfurt, Germany. To appear.
- [16] Wang, L., Sy, A., Liu, L. and Piech, C. 2017. "Deep knowledge tracing on programming exercises." *In Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pp. 201-204.
- [17] Xiong, X., Zhao, S., Van Inwegen, E. G., & Beck, J. E. 2016. Going deeper with deep knowledge tracing. *International Educational Data Mining Society*.
- [18] Zhang, H., and Litman, D. 2019. "Co-attention based neural network for source-dependent essay scoring." *arXiv preprint arXiv:1908.01993*.

exposed to LearnSphere from these past events, although we will have some tutorial activities included for new attendees as well. This workshop builds off a successful LAK 2018 Tutorial, workshop at AIED/EDM 2017, and workshops at EDM 2019.

2. ORGANIZATIONAL DETAILS

2.1 Type of Event

Workshop

2.2 Proposed Schedule

Table 1. Proposed Half-day Schedule

Time	Item
8:30	Introductions
9:00	Tigris workflow tool (Lecture & Demos)
10:10	Coffee Break
10:20	Hands on: Build custom analysis workflows using existing Tigris components
11:20	5-minute participant talks about proposed or created workflows
12:00	Closing / High-level Discussion

2.3 Type of Participation

Mixed participation will be through submission of reviewed abstracts, invited guests, and open registration. For participants who have accepted abstracts or are invited by the workshop committee, we have allocated approximately \$15,000 from our grant funding to cover registration and some travel costs of select participants based on quality of submissions, attract students and junior faculty, and a goal to create a diverse set of participants.

2.4 Activities

Activities will include presentations from workshop organizers, invited guests, and short presentations from accepted abstract presenters. Hands on sessions will include demos and group work towards implementing analytics.

2.5 Expected Numbers

We expect 15-20 participants based on previous workshops.

2.6 Activities to Recruit Attendees

We will create a website to announce the workshop and method of submitting abstracts. The Learning Analytics, Educational Data Mining, and LearnLab mailing lists will be used to direct potential attendees to the workshop website. In addition, we will include a number of invited guests. Both accepted submissions and invited guests will have the chance to receive funding to attend.

2.7 Required Equipment

Projector and screen will be required by organizers. Attendees will need to bring laptops and will need adequate internet connectivity.

3. OBJECTIVES AND OUTCOMES

Broadly, this workshop offers those in the Learning Analytics community an exposure to LearnSphere as a community-based

infrastructure for educational data and analysis tools. In opening lectures, the organizers will discuss the way LearnSphere connects data silos across universities and its unique capabilities for sharing data, models, analysis workflows, and visualizations while maintaining confidentiality.

More specifically, we propose to focus on attracting, integrating, and discussing researcher contributions to Tigris, the web-based workflow authoring and sharing tool. Workshop submissions in the form of abstracts will involve a brief description of an analysis pipeline relevant to modeling educational data as well as accompanying code. Prior to the workshop itself, the organizers will coordinate with authors of accepted submissions to integrate their code into Tigris. A significant portion of the workshop will be dedicated to hands-on exploration of custom workflows and workflow modules within Tigris. Authors of accepted submissions will present their analysis pipelines, and everyone attending the workshop will be able to access those analysis pipelines within Tigris to a variety of freely available educational datasets available from LearnSphere. The goal is to generate -- for each workflow component contribution in the workshop -- a publishable workshop paper that describes the outcomes of openly sharing the analysis with the research community.

Finally, workshop attendees will discuss bottlenecks that remain toward our goal of a unified repository. We will also brainstorm possible solutions. Our goal is to create the building blocks to allow groups of researchers to integrate their data with other researchers we can advance the learning sciences as harnessing and sharing big data has done for other fields.

4. REFERENCES

- [1] Jo, Y., Tomar, G., Ferschke, O., Rosé, C. P., & Gašević, D. (2016, April). Pipeline for expediting learning analytics and student support from data in social learning. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge* (pp. 542-543). ACM.
- [2] Koedinger, K. R., Corbett, A. T., & Perfetti, C. (2010). The knowledge-learning-instruction (KLI) framework: Toward bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*.
- [3] Stamper, J., Koedinger, K.R., Baker, R., Skogsholm, A., Leber, B., Demi, S., Yu, S., Spencer, D. (2011) Managing the Educational Dataset Lifecycle with DataShop. In Kay, J., Bull, S. and Biswas, G. (eds). *Proceeding of the 15th International Conference on Artificial Intelligence in Education* (AIED2011).
- [4] Veeramachaneni, K., Halawa, S., Dernoncourt, F., O'Reilly, U. M., Taylor, C., & Do, C. (2014). Moocdb: Developing standards and systems to support MOOC data science. *arXiv preprint*. arXiv:1406.2015.

Educational Data Mining in Computer Science Education (CSEDM) Workshop

Thomas W. Price
North Carolina State University
twprice@ncsu.edu

Peter Brusilovsky
University of Pittsburgh
peterb@pitt.edu

Sharon I-Han
Arizona State University
sharon.hsiao@asu.edu

Kenneth R. Koedinger
Carnegie Mellon University
koedinger@cs.cmu.edu

Yang Shi
North Carolina State University
yshi26@ncsu.edu

ABSTRACT

There is a growing community of researchers at the intersection of data mining, AI and computing education research. The objective of the CSEDM workshop is to facilitate a discussion among this research community, with a focus on how data mining can be uniquely applied in computing education research. For example, what new techniques are needed to analyze program code and CS log data? How do results from CS education inform our analysis of this data? The workshop is meant to be an interdisciplinary event at the intersection of EDM and Computing Education Research. Researchers, faculty and students are encouraged to share their AI- and data-driven approaches, methodologies and experiences where data is transforming the way students learn Computer Science (CS) skills. This full-day workshop will feature a panel, paper presentations, discussions to promote collaboration, and a kick-off of the 2nd CSEDM Data Challenge.

Keywords

Computer Science Education, Educational Data Mining, AI in Education, Learning Analytics

1. WORKSHOP GOALS

Computing is an increasingly fundamental skill for students across disciplines. It enables them to solve complex, real and challenging problems and make a positive impact in the world. Yet, the field of computing education is still facing a range of problems from high failure and attrition rates, to challenges training and recruiting teachers, to the underrepresentation of women and students of color.

Advanced learning technologies, which use data and AI to improve student learning outcomes, have the potential to address these problems. However, the domain of CS education presents novel challenges for applying these techniques.

CS presents domain-specific challenges, such as helping students effectively use tools like compilers and debuggers, and supporting complex, open-ended problems with many possible solutions. CS also presents unique opportunities for developing learning technologies, such as abundant and rich log data, including code traces that capture each detail of how students' solutions evolved over time.

These domain-specific challenge and opportunities suggest the need for a specialized community of researchers, working at the intersection of AI, data-mining and computing education research. The goal of this 4th Educational Data Mining for Computer Science Education (CSEDM) workshop¹ is to bring this community together to share insights for how to support and understand learning in the domain of CS using data. This field is nascent but growing, with researching in computing education increasingly using data analysis approaches, and researchers in the EDM community increasing studying CS datasets. This workshop will help these researchers learn from each other, and develop the growing sub-field of CSEDM.

The CSEDM workshop is co-organized with CS-SPLICE (cssplice.org), an NSF-funded organization that seeks to build infrastructure for intelligent learning content in CS education. The workshop will build on three successful prior CSEDM workshops at: 1) the International Educational Data Mining Conference (EDM) in 2018², 2) the International Learning Analytics and Knowledge Conference (LAK) in 2019³, and 3) and the International Conference on AI in Education (AIED) in 2019⁴. Each were fruitful and well-attended. We hope to keep our momentum with a 4th CSEDM Workshop, returning to EDM in 2020. We also build on the success of 5 prior SPLICE workshops at CS education conferences (ACM SIGCSE, ACM ICER).

1.1 Relevant Topics

The workshop encourages contributions from the following topics of interest:

- Predictive and descriptive modelling for CS courses

¹sites.google.com/ncsu.edu/csedm-ws-edm-2020/

²sites.google.com/asu.edu/csedm-ws-edm-2018/

³sites.google.com/asu.edu/csedm-ws-lak-2019/

⁴sites.google.com/asu.edu/csedm-ws-aied-2019/

Thomas Price, Peter Brusilovsky, Sharon Hsiao, Kenneth Koedinger and Yang Shi "Educational Data Mining in Computer Science Education (CSEDM) Workshop" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 826 - 828

- Adaptation and personalization within CS learning environments
- Intelligent support for collaborative CS problem solving
- Machine learning approaches to analyze massive CS datasets and courses
- Online learning environments for CS: implementation, design and best practices
- Multimodal learning analytics and combination of student data sources in CS Education
- Affective, emotional and motivational aspects related to CS learning
- Adaptive feedback, adaptive testing for CS learning
- Discourse and dialogue research related to classroom, online, collaborative, or one-on-one learning of CS
- Peer-review, peer-grading and peer-feedback in CS
- Teaching approaches using AI tools
- Visual Learning Analytics and Dashboards for CS
- Network Analysis for programming learning environments
- Self-Regulated learning for CS environments
- Writing and syntax analysis for programming design learning
- Natural Language Processing for CS forums and discussions
- Analysis of programming design and trajectory paths
- Recommender systems and in-course recommendations for CS learning

We will invite researchers who are interested in further exploring, contributing, collaborating and developing data- and AI-driven techniques for building educational tools for Computer Science to submit paper on any of these topics.

2. WORKSHOP ORGANIZATION

The workshop will be organized by a team with a history of CSEDM research:

Thomas Price is an Assistant Professor of Computer Science at North Carolina State University. His primary research goal is to develop learning environments that automatically support students through AI and data-driven help features. His work has focused on the domain of computing education, where he has developed techniques for automatically generating programming hints and feedback for students in real-time by leveraging student data. He has helped organized a number of efforts at the intersection of AIED, Data Mining and CS Education, including the CS-SPLICE working group on programming snapshot representation and prior CSEDM and CS-SPLICE workshops.

Peter Brusilovsky is a Professor of Information Science and Intelligent Systems at the University of Pittsburgh, where he also directs Personalized Adaptive Web Systems (PAWS) lab. He has been working in the field of adaptive educational systems, user modeling, and intelligent user interfaces for more than 30 years. He published numerous papers and

edited several books on adaptive hypermedia and the adaptive Web. He is a founder of CS-SPLICE and has advanced research and infrastructure for CSEDM.

Sharon I-Han Hsiao is an Assistant Professor at the School of Computing, Informatics & Decision Systems Engineering in Arizona State University. Her research lies in the intersections of Informatics & Computational Technologies for Learning with a focus on Intelligent Tutoring Systems, Computer Science Education, Adaptive Educational Systems, Open User Modeling, Data Sciences, Visualization, Social Computing, and Learning Technologies.

Ken Koedinger is Professor of Human-Computer Interaction and Psychology at Carnegie Mellon. He explores how people think and learn by developing and studying technology-enhanced learning. He leads the LearnSphere effort (learnsphere.org), which integrates learning data and analytics across multiple resources. And he directs LearnLab (learnlab.org), which started with 10 years of National Science Foundation funding and is now the scientific arm of CMU's Simon Initiative (cmu.edu/simon). He is also a founder of CS-SPLICE.

Yang Shi is a PhD student at North Carolina State University. His research focuses on developing data-driven methods for representing program code to enhance the ability of intelligent learning environment to support students and model their learning. Yang's research interest includes CSEDM, Automatic Hint Generation, Programming Language Processing, Software Representations, Software Analysis and Deep Learning.

2.1 Program Committee

The 4th CSEDM Workshop's program committee includes:

- Austin Cory Bart (University of Delaware, USA)
- Barbara Ericson (University of Michigan, USA)
- Petri Ihantola (University of Helsinki, Finland)
- Juho Leinonen (University of Helsinki, Finland)
- Cliff Shaffer (Virginia Tech University, USA)
- Alan Smeaton (Dublin City University, Ireland)
- Sergey Sosnovsky (Utrecht University, Netherlands)
- John Stamper (Carnegie Mellon University, USA)
- Michael Yudelson (ACT)

3. CALL FOR PARTICIPATION

We will solicit two types of research contributions:

4-8 page Research Papers: Original, unpublished work or work-in-progress, addressing any of the topics of interest above.

2-4 page Presentation Abstracts: Researchers will present their work at CSEDM in a conversational format. Presentations might include:

- Descriptions of shareable Computer Science (CS) datasets

- Descriptions of data mining / analytics approaches applied to specifically Computer Science datasets
- Descriptions of tools or programming environments that use/produce data
- Case studies of collaboration where reproducible practices were used to integrate or compose two or more data analysis tools from different teams
- Descriptions of infrastructures that could collect and integrate data from multiple learning tools (e.g. forum posts, LMS activity and programming data)

4. WORKSHOP ACTIVITIES

The workshop will be a **half day** workshop, held online on July 10th, 2020. It will primarily consist of paper presentations, discussions to facilitate collaboration, and a kickoff of the 2nd CSEDM Data Challenge. The full schedule can be found at the workshop website: sites.google.com/ncsu.edu/csedm-ws-edm-2020

4.1 2nd CSEDM Data Challenge – Kickoff

A unique aspect of the CSEDM workshops is the CSEDM Data Challenge. The goal of this challenge is to bring researchers together to tackle a common data mining task that is specific to CS Education. We are building on the success of the first CSEDM Data Challenge⁵. The first challenge focused on the task of modeling students' programming knowledge in order to predict their performance on future tasks. Researchers competed to build the best predictive model, using a common dataset. This year, we will use the CSEDM workshop to kick off a second Data Challenge. Our goal is to use the workshop as a space to build researcher interest in the challenge, introduce the datasets to be used, and get input from the community on their goals for the challenge.

5. SOLICITATION PLAN

Building on our growing network of contributors to prior workshops, we intend to solicit participation on the workshop through the following mailing lists and research networks:

- ACM's Special Interest Group on Computer Science Education (SIGCSE)
- Computer Science Education (CSED) research list (from the ICER community)
- European Association of Technology-Enhanced Learning (EATEL) community
- User Modeling (UM) mailing list
- Asia-Pacific Society for Computers in Education (AP-SCE) community
- PSLC community list
- Relevant EU project consortia
- The International Educational Data Mining Society
- The Society for Learning Analytics Research (SoLAR)

⁵github.com/thomaswp/CSEDM2019-Data-Challenge

Causal Inference in Educational Data Mining

Half-day workshop

EDM 2020

Fully Virtual

1. OVERVIEW

The goal in crafting intelligent tutoring systems, educational games, MOOCs, and other computerized learning tools, is to improve student learning. To that end, EDM research typically focuses on methods to identify, measure, and predict learner behaviors or outcomes. Causal research seeks to estimate the impacts of different factors on these behaviors or outcomes—not only predicting who will wheel-spin, experience frustration, or successfully learn a new skill, say, but determining causes these? Causality lies at the heart of both learning science, which seeks to understand how inputs in an educational system affect the system’s outputs, and of policy, which seeks to design educational systems that improve learning.

The field of causal inference, which spans statistics, philosophy, economics, computer science, and other more traditional academic disciplines, has itself experienced rapid and exciting developments in the recent past. The new science of causality encompasses new ways of estimating effects under challenging circumstances, such as possible confounding, but also new questions—how do impacts vary between learners? What mechanisms drive causal effects? How may we construct optimal individualized policies for specific learners?

This workshop is intended to raise awareness of the ubiquity and importance of causal questions in EDM, some of the exciting methods available to address those questions, and some of the open questions of causal inference in EDM. It will include invited discussions of ongoing projects addressing causal questions, and short talks about relevant work in progress, including work in any stage of development.

Lastly, the workshop will give an opportunity for EDM researchers to submit open problem related to causality in EDM research, in an exercise motivated by the Quantitative Methods Program seminar at the University of Michigan Institute for Social Research. In five minute presentations, researchers will briefly present problems they have encountered

in research, or that they just think are interesting, but that they do not yet know how to solve. Each presentation will be followed by an open-ended discussion among the workshop participants, hopefully suggesting ways to solve, or at least better refine the problem. This sub-workshop will hopefully give the presenting researchers constructive suggestions, and spur collaborations.

In general, the workshop will be organized to stimulate discussion among participants, including, hopefully, constructive suggestions for open problems.

2. TOPICS OF INTEREST

We will solicit work on topics including, but not limited to:

- A/B Testing
- Graphical causal models/Bayesian networks
- Analyzing data from randomized experiments
- Multi-armed bandits
- Investigations of causal mechanism/mediation analysis
- Estimating EDM program impacts
- Identifying and predicting differential effects
- Connections between machine learning and causal inference
- Dynamic treatment regimes
- Principal stratification
- Causal inference in EDM without randomization

3. ORGANIZERS

- Adam Sales (University of Texas, Austin/Worcester Polytechnic Institute)
- Stephen Fancsali (Carnegie Learning)
- Anthony Botelho (Worcester Polytechnic Institute)
- Joseph Jay Williams (University of Toronto)
- Neil Heffernan (Worcester Polytechnic Institute)

Adam Sales "Workshop: Causal Inference in Educational Data Mining" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 829 - 830

0:00-0:30	Introduction: Causal questions in EDM (Adam Sales)
0:30-1:30	Two long (20 minute) talks, with discussion
1:30-2:15	Three talks describing work in progress (10 minutes each), with discussion
2:15-3:15	Open problems workshop
3:15-3:30	concluding discussion

Table 1: Preliminary Schedule

4. IMPORTANT DATES

- Submission deadline (Extended): Monday, June 15, 2020
- Acceptance notification (Extended): Monday, June 29, 2020

Reviewing process: The workshop organizers will review papers, alongside external reviewers whose expertise is appropriate for the submissions. These may include Anna Rafferty (Carleton College), Thanaporn March Patikorn (Worcester Polytechnic Institute), Peter Schaldenbrand (Carnegie Mellon University) and others.

FATED: Fairness, Accountability, and Transparency in Educational Data (Mining)

Nigel Bosch
University of Illinois at
Urbana–Champaign
pnb@illinois.edu

Christopher Brooks
The University of Michigan
brooksch@umich.edu

Shayan Doroudi
University of California, Irvine
doroudis@uci.edu

Josh Gardner
University of Washington
jpgard@cs.washington.edu

Kenneth Holstein
Carnegie Mellon University
kjholste@cs.cmu.edu

Andrew S. Lan
University of Massachusetts
Amherst
andrewlan@cs.umass.edu

Collin Lynch
North Carolina State
University
cflynch@ncsu.edu

Beverly Park Woolf
University of Massachusetts
Amherst
bev@cs.umass.edu

Mykola Pechenizkiy
Eindhoven University of
Technology
m.pechenizkiy@tue.nl

Steven Ritter
Carnegie Learning, Inc.
sritter@carnegielearning.com

Jill-Jënn Vie
Inria Lille
jill-jenn.vie@inria.fr

Renzhe Yu
University of California, Irvine
renzhey@uci.edu

ABSTRACT

This document outlines a proposed full-day workshop focused on the intersection of fairness, accountability, transparency, and educational data mining (EDM). The workshop aims to provide a multidisciplinary perspective on fairness-related work from both “sides” of the EDM community (*education* and *data mining*) along with other relevant fields (human–computer interaction, machine learning, etc.). Our workshop aims to be an inclusive opportunity for EDM researchers to learn about an emerging field, as well as to define a research agenda for this area of critical importance to the field.

1. BACKGROUND

As data-driven algorithms are increasingly relied upon to shape user experiences, deliver content, and make decisions across a variety of domains, concerns have grown around the fairness, equity, accountability, transparency, and inclusivity of algorithmic systems and the broader pipelines within which they exist (e.g., training data, human decisions made using algorithmic outputs). The field of educational data mining, being concerned with the use of data about human

subjects for the purposes of studying educational processes and improving learning outcomes, is intimately tied to these concerns about fairness, accountability, and transparency.

On the other hand, equity is at the heart of the development goals of education across the globe, given the personal, economic and social benefits of education [26]. Accordingly, decades of education research have been devoted to understanding existing inequities and finding ways to address them [11, 5]. Concerns around fairness in data mining used in educational contexts must therefore be viewed in the broader context of educational concerns around equity. In fact, early work on fairness in educational testing dating back to the 1960s preempted many contemporary definitions of fairness that have emerged in the machine learning (ML) literature [16].

Because this cluster of topics¹ is still an emerging field both within the EDM community and the broader field of data mining, we believe that it is the responsibility of EDM researchers to be at the forefront of defining the future research of such a field, and to incorporate FATE-related inquiries into their work. Towards this ends, this document describes a full-day workshop to explore the intersection of fairness research (broadly construed) and educational data mining. We propose a “bidirectional” approach, wherein the workshop focuses both on synthesizing past research as well as defining an agenda for future work in the field. Our or-

Nigel Bosch, Christopher Brooks, Shayan Doroudi, Josh Gardner, Kenneth Holstein and Renzhe Yu “FATED: Fairness, Accountability, and Transparency in Educational Data (Mining)” In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 831 - 834

¹As an emerging subfield, the area of Fairness, Accountability, Transparency (and often also Ethics) does not have a single agreed-upon name; in this document, we use the term “FATE” to refer to these topics jointly.

ganizing committee draws from a broad set of disciplinary backgrounds, including machine learning, human-computer interaction (HCI), and education, and we intend the workshop to serve as a unifying forum for a variety of research and disciplinary perspectives.

We believe that the proposed workshop will provide a firm foundation for participants to conduct future impactful FATE-related EDM research by providing both a foundation in past work and a well-defined set of research goals for future work to address. Furthermore, we believe that this workshop will serve to advance the EDM2020 theme, “Improving Learning Outcomes for All Learners.”

2. PRIOR WORK

This section does not attempt to provide an exhaustive list of work relevant to the workshop. Instead, it is intended to provide a sense of the work in this area which we hope to introduce to participants and build upon in our intended outcomes.

2.1 FATE in Data Mining and ML

Over the past decade, a rich subfield of fairness research has emerged within the data mining and machine learning communities. Research has investigated the (un)fairness of data mining and machine learning algorithms applied to several different applications, including image classification [8] and natural language processing [6]. Additionally, there is a rich and ever-growing body of literature on methods and metrics for fair data mining and machine learning [3]. This has also included research into the biases inherent to data itself, as our workshop title implies. Finally, work has also begun to explore the relationship between the utility, fairness, and privacy of statistical models [19, 18], all of which are critical to the success of educational models in certain contexts [15].

While several conferences and other publication venues have emerged to share research related to fairness² as well as subgroup-specific venues for advancing such work within specific subfields³, EDM has not yet offered a workshop to address such topics directly.⁴

2.2 Equity in Education

Educational equity and achievement gaps are topics that have been studied by education researchers and have been of paramount importance to the practice of education for generations [11]. Empirical research in the past few decades have found consistent evidence of systematic gaps in educational opportunities and outcomes by socioeconomic status, immigration status and gender [5]. Of relevance to educational data mining, [16] surveys several notions of testing-related fairness in education and employment, demonstrating that several recent conceptions of fairness were anticipated in prior works as early as the 1960s. In the context of educational technology, a number of studies have shown

²ACM FAccT <https://facctconference.org/>,
AAAI/ACM AIES <https://www.aies-conference.com/2020/>

³See e.g. Fair ML for Health <https://www.fairmlforhealth.com/>

⁴For a related effort, see <https://sites.google.com/view/fairlak>.

that even with equal access, more privileged students may disproportionately benefit from technologies than less privileged or marginalized students, often due to socio-cultural factors [24]. Relatedly, recent work on online learning has found that usage patterns vary by demographic groups in diverse global learning environments [14] and that identity-based interventions have disparate impact in adaptive learning environments [7, 21]. Work on fairness in EDM should situate itself in the context of this broader work on educational equity. One of the goals of this workshop will be to bring voices from educational researchers to help ensure a desirable path forward for FATE in EDM.

2.3 Fairness in EDM

Within the EDM community, some earlier research has investigated the generalizability of student models to new student populations and/or learning contexts (aka *external validity*) but found mixed results [2, 25]. While this investigation remains a critical research direction of EDM research [1], the fairness perspective speaks more to the *internal validity* of data mining models. Towards this end, recent research in educational data mining and learning analytics has evaluated the fairness of on-time graduation models [17], mastery learning algorithms in tutoring systems [12], dropout models in MOOCs [13], and the effects of perceived AI (un)fairness in college admissions [23]. Combined with the influence of socio-cultural factors as mentioned above, these works collectively suggest that the ways in which fairness-related issues intersect with the methods and goals of educational data mining systems are complex, multidimensional, and in need of further research.

2.4 Transparency in EDM

Data mining is often integrated into educational systems to automate or enhance decision-making processes that might otherwise be performed by humans, such as homework grading [10] or personalizing learning content [4]. Automated decision-making may affect students in large ways (e.g., automatic homework grading assigning a failing grade) as well as relatively minor ways (e.g., selecting the next topic of study), though even minor effects may accumulate into large outcomes. For students, teachers, and other stakeholders to trust decisions made by data mining algorithms, it is essential that those algorithms provide transparent explanations of their decision-making process at an appropriate level [20].

However, even simple models such as linear regression with a few variables or Bayesian knowledge tracing [9] can be difficult for many users to understand [22]. Previous work has explored the possibility of creating explanations specifically for Bayesian knowledge tracing [27], and found a great deal of heterogeneity in the level of explanation needed across users (e.g., students), as well as the desire different users had for an explanation. Given that the difficulty of creating and selecting an appropriate explanation is enhanced by factors such as age, native language, and neurodiversity that vary across individuals, this is a large area of potential research that we propose to discuss and define with respect to educational data mining in this workshop.

3. WORKSHOP STRUCTURE

We propose a “bidirectional” workshop structure, with a backward-looking component focused on synthesis of both

historical and state-of-the-art work, as well as a forward-looking component focused on defining a research agenda for the field.

3.1 Backward-Looking Activities

The backward-looking component will be focused on bringing together a comprehensive and multidisciplinary view of how to measure, identify, and correct for lack of fairness in fields relevant to educational data mining research. Given that there has been an explosion of work on fairness in machine learning over the past few years, it can be difficult for researchers to begin working in this literature without getting lost. Rather than providing a broad tutorial of all definitions and approaches to fairness, we aim to provide an introduction that more thoroughly examines a smaller set of approaches and recent discussions relevant to the field of educational data mining. We intend for this section to include a mix of presenter-led synthesis presentations as well as participant-led presentations (invited or submitted) providing perspectives from specific research studies. The goal for this component is to provide a firm foundation in the state of the field for interested researchers, including those new to fairness-related research.

3.2 Forward-Looking Activities

The forward-looking component will be focused on identifying measures of success, key open problems, and a research agenda for the emerging work in fairness for educational data mining. This section will be largely participant-driven. One possible activity for this section is a reverse assumptions activity (a method from HCI and UX design), wherein participants iteratively construct “positive” and “negative” design fictions for fairness-related research in EDM. In this way, we can not only pinpoint concrete ways in which unfairness can creep into well-intentioned solutions (negative design fictions), but also identify solutions that can mitigate concerns around fairness and equity (positive design fictions). Finally, we aim to conduct work on a collaborative document providing a taxonomy of fairness-related work in EDM and a list of open problems to focus efforts in the field. We will facilitate these activities through a series of brainstorming activities that will consist of individual idea generation, synthesis and refinement of ideas via group discussion, and subsequent group work toward formalizing and expanding ideas through groups formed around interest in specific ideas.

4. INTENDED OUTCOMES

Our intended outcomes for this workshop are twofold. First, we aim to provide a foundation for workshop participants in prior fairness-related work relevant to the EDM community. The resources used for this component of the workshop will be made available open-source, and we hope that this will support the larger EDM community and provide a catalyst for future learning. Second, we aim to provide a set of concrete community-created resources for fairness-related work in EDM. These include the results of our design fictions activities, definitions of open problems in the field, and a taxonomy to direct future research efforts in this burgeoning subfield. These outputs will be disseminated to the broader educational data mining community. Ultimately, the long-term goal of this workshop is not to create a niche commu-

nity of EDM researchers interested in fairness, accountability, and transparency, but to encourage all researchers and practitioners in the EDM community to think about how to ensure FATE in the work they do.

5. REFERENCES

- [1] R. S. Baker. Challenges for the future of educational data mining: The baker learning analytics prizes. *JEDM| Journal of Educational Data Mining*, 11(1):1–17, 2019.
- [2] R. S. Baker and S. M. Gowda. An analysis of the differences in the frequency of students’ disengagement in urban, rural, and suburban high schools. In *Proceedings of the 3rd International Conference on Educational Data Mining (EDM)*, pages 11–20, 2010.
- [3] S. Barocas, M. Hardt, and A. Narayanan. *Fairness in machine learning*. <https://fairmlbook.org/>, 2019.
- [4] J. Bassen, B. Balaji, M. Schaarschmidt, J. Painter, D. Zimmaro, A. Games, E. Fast, C. Thille, and J. Mitchell. Reinforcement learning for the scheduling of online learning activities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020.
- [5] J. Blanden. Education and inequality. In S. Bradley and C. Green, editors, *The Economics of Education (Second Edition)*, pages 119 – 131. Academic Press, second edition, 2020.
- [6] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.
- [7] C. Brooks, J. Gardner, and K. Chen. How gender cues in educational video impact participation and retention. International Society of the Learning Sciences, Inc.[ISLS]., 2018.
- [8] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91, 2018.
- [9] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1995.
- [10] S. A. Crossley, L. K. Allen, E. L. Snow, and D. S. McNamara. Incorporating learning characteristics into automatic essay scoring models: What individual differences and linguistic features tell us about writing quality. *Journal of Educational Data Mining*, 8(2):1–19, 2016.
- [11] L. Darling-Hammond. *The flat world and education: How America’s commitment to equity will determine our future*. Teachers College Press, 2015.
- [12] S. Doroudi and E. Brunskill. Fairer but not fair enough on the equitability of knowledge tracing. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 335–339, 2019.
- [13] J. Gardner, C. Brooks, and R. Baker. Evaluating the fairness of predictive student models through slicing analysis. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 225–234, 2019.

- [14] J. D. Hansen and J. Reich. Democratizing education? examining access and usage patterns in massive open online courses. *Science*, 350(6265):1245–1248, 2015.
- [15] K. Holstein, J. Wortman Vaughan, H. Daumé III, M. Dudik, and H. Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2019.
- [16] B. Hutchinson and M. Mitchell. 50 years of test (un) fairness: Lessons for machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 49–58, 2019.
- [17] S. Hutt, M. Gardner, A. L. Duckworth, and S. K. D’Mello. Evaluating fairness and generalizability in models predicting on-time graduation from college applications. *International Educational Data Mining Society*, 2019.
- [18] N. Kallus, X. Mao, and A. Zhou. Assessing algorithmic fairness with unobserved protected class using data combination. *arXiv preprint arXiv:1906.00285*, 2019.
- [19] N. Kallus and A. Zhou. Residual unfairness in fair machine learning from prejudiced data. *arXiv preprint arXiv:1806.02887*, 2018.
- [20] R. F. Kizilcec. How much information? effects of transparency on trust in an algorithmic interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2390–2395, 2016.
- [21] R. F. Kizilcec and A. J. Saltarelli. Psychologically inclusive design: Cues impact women’s participation in stem education. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–10, 2019.
- [22] Z. C. Lipton. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.
- [23] F. Marcinkowski, K. Kieslich, C. Starke, and M. Lünich. Implications of ai (un-)fairness in higher education admissions: The effects of perceived ai (un-)fairness on exit, voice and organizational reputation. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAT* ’20*, page 122–130, New York, NY, USA, 2020. Association for Computing Machinery.
- [24] J. Reich and M. Ito. From good intentions to real outcomes: Equity by design in learning technologies. *Digital Media and Learning Research Hub*, 2017.
- [25] B. Samei, A. M. Olney, S. Kelly, M. Nystrand, S. D’Mello, N. Blanchard, and A. Graesser. Modeling Classroom Discourse: Do Models that Predict Dialogic Instruction Properties Generalize across Populations? 2015.
- [26] UNESCO. Education 2030: Incheon Declaration and Framework for Action for the implementation of Sustainable Development Goal 4: Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all. 2016.
- [27] H. H. I. Zhou, T; Sheng. Assessing post-hoc explainability of the bkt algorithm. *AIES*, 2020.